AT THE INTERSECTION OF DIFFERENTIAL EQUATIONS AND OPTIMIZATION: INVERSE PROBLEMS, PATH PLANNING AND KRYLOV SUBSPACES

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Marc Aurèle Tiberius Gilles

May 2019

© 2019 Marc Aurèle Tiberius Gilles
ALL RIGHTS RESERVED

AT THE INTERSECTION OF DIFFERENTIAL EQUATIONS AND OPTIMIZATION: INVERSE PROBLEMS, PATH PLANNING AND KRYLOV SUBSPACES

Marc Aurèle Tiberius Gilles, Ph.D.

Cornell University 2019

Four problems at the intersection of optimization and partial differential equations are presented. First, a problem in remote sensing of the marine atmospheric boundary layer is discussed. A method that exploits the low-rank structure of the electromagnetic field is used to infer the refractive index profile of the lower atmosphere. The second problem is concerned with 3D X-ray imaging of large objects at nanometer scale resolution. A massively parallel optimization method is used to perform the reconstruction from measurements of an object outside of the depth of focus. The third problem presents a path planning problem where an evader is choosing his trajectory to hinder the surveillance of an observer. An algorithm to compute optimal strategies using ideas from convex optimization, game theory and optimal control is described. The final chapter presents a practical framework to apply Krylov subspace methods to differential operators.

BIOGRAPHICAL SKETCH

Marc Aurèle Gilles was born to Marie Ollivot Gilles and Erard Gilles in Paris, France on April 21st, 1992. He grew up in Saint-Malo, a small city on the North-west coast of France. After graduating from high school, he moved to New Jersey and attended Raritan Valley Community College and Rutgers University where he studied mathematics and economics. He started his doctorate degree at Cornell University in August 2014. In the summer of 2017, he was an NSF intern at Argonne National Laboratory and in the summer of 2018, he was a research intern at the Facebook Reality Labs where he returned as a research scientist after graduating.

To my mother, Marie Ollivot Gilles.

ACKNOWLEDGEMENTS

Thank you to my advisor Alex Townsend for his benevolent guidance, his contagious optimism and his unwavering support in doing whatever interested me. I will truly miss leaving his office enthusiastic and curious every week.

Thank you to my committee members and coauthors Alex Vladimirsky, Christopher Earls, David Bindel, and Stefan Wild for their time, helpful guidance and patience while working with me. I am very grateful that they gave me the opportunity to work on fascinating projects from which I learned most of what I know, and from which I derived a lot of research excitement. Chapter 2 of this dissertation is joint work with Christopher Earls and David Bindel, chapter 3 is joint work with Youssef Nashed, Ming Du, Christopher Jacobsen, and Stefan Wild, chapter 4 is joint work with Alex Vladimirsky, and chapter 5 is joint work with Alex Townsend.

I also want to thank all of the people who helped me get to and through graduate school: my mother, father, Isabelle, my brothers and sisters, Sara, and my undergraduate mentor Shabnam Beheshti.

Finally, thanks to my Ithaca friends who made these years not only research-fun, but also actually fun: David, Kun, Matt, Mateo, Sharon, Zach, Kyle, Andrew, Heather, Tianyi, Dan, and the rest of the CAM students.

This work was supported by the Center for Applied Mathematics, the National Science Foundation through grant 1645445, grant DMS-1738010 and the Mathematical Science Graduate Internship program, the Office for Naval Research through grant N00014-16-1-2077, and through teaching assistantships in Cornell's Departments of Mathematics, Computer Science and Information Science.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

This dissertation consists of four problems shallowly linked by the fact that they each contain an optimization problem coupled with a differential equation. The number of problems that fall into this category is enormous which is why, despite this link, these four projects are mostly unrelated. However, all of them involve two aspects that I enjoy about applied mathematics: efficient numerics and careful algorithmic design.

1.1 Inverse problems

The first two chapters of this dissertation treat inverse problems. An inverse problem appears when one wants to infer from a set of observations the causal factor which produced them. These problems arise in a wealth of applications including remote sensing, computer vision, tomography, and astronomy. Inverse problems are associated with a *forward problem*: a model of the process that produces the data [160]. Often, this forward model takes the form of a differential equation. For example,

- In chapter 2, the goal is to infer the index of refraction of the environment. The forward model is a Helmholtz equation, which dictates the propagation of electromagnetic waves, and the observations are radar measurements.
- In chapter 3, the goal is to infer the material properties and shape of an object that is being imaged through an X-ray. The forward model is the

multi-slice propagation model, and the observations are diffractive patterns.

Inverse problems are commonly solved by answering some version of the question: determine the causal factor that, through the forward model, produces the measurements which most resemble the data. In mathematical term, this is written as an optimization problem of the form:

$$\min_{x \in S} \|f(x) - y\|. \tag{1.1}$$

Here f represents the forward model, x the causal factor, y the observations, S a set that can be interpreted as the physical bounds on x, and the norm used is problem dependent. Although the number of inverse problems is gigantic and each is different from the next, they have some frequent pitfalls:

1. They are typically ill-posed. That is, a small change in the measurements can induce a large change in the reconstruction. This is particularly worrying given that any physical measurement is contaminated by noise. A common way to help alleviate this issue is the use of *regularization*: adding extra information we have about the desired solution into the optimization problem.
2. The forward model might be expensive to evaluate. For example, when it requires the solution of a differential equation or when the forward model is very high dimensional. This is especially problematic because inverse problems usually require many evaluations of the forward model, and thus efficient numerics in the implementation of the forward model is crucial.

3. The forward model might be a highly non-linear function of the causal factor, causing the objective function to have many local minima. This results in the optimization problem in eq. (1.1) being difficult in the sense that it requires a lot of computational power.

Hence, when attempting to solve an inverse problem, one typically has to design an algorithm to carefully balance multiple needs:

1. We would like a forward model that is an accurate model of the real world but we also need it to be simple enough that it can be cheaply evaluated many times (see chapter 3).
2. The optimization problem needs to lead to a desirable solution, but we also want the optimization problem to be easy to solve (see chapter 2). This can limit the type of regularization that is used.

1.1.1 Remote sensing of the atmosphere

Chapter 2 presents an inverse problem in remote sensing: inferring the refractive index profile of the atmosphere from radar measurements. The main difficulty in solving this inverse problem comes from the poor behavior of the objective function. Indeed, the minimization problem in eq. (1.1) is hard to solve due to the fact that the function has thousands of local minima. Most of these minima look promising to a typical optimization algorithm, and an exhaustive search of all of them is computationally intractable. The solution proposed in chapter 2 is to use a different objective function that leverages the observation that, in the region of interest, vertical slices of the electromagnetic (EM)

field are of low-rank. This low-rank structure is inherited by solutions of the Helmholtz equation, which governs the propagation of EM waves. The solution of the Helmholtz equation may be written as a sum of modes corresponding to eigenfunctions of a Sturm–Liouville eigenvalue problem associated with a specific refractive index profile [81]. Most of these modes decay exponentially fast away from the source of the EM wave, and thus only a few propagate downrange, giving rise to the low-rank structure. Instead of the typical objective function that answers the question "find the atmosphere which produces radar measurements that most resemble the observations", we employ one which answers "find the atmosphere whose associated set of eigenfunctions best fits the observations". We find that this approach leads to a much better behaved objective function, allowing us to characterize the atmospheric conditions in real-time.

1.1.2 Nanoscopic 3D imaging

Chapter 3 presents an inverse problem in 3D imaging: reconstruct a nanometer scale resolution 3D image of a large, complicated object using X-ray tomography. Objects imaged through X-ray typically fit in the "depth of focus" of the optical system. In this region, the interaction of the X-ray wave with the object it traverses can be well approximated as the interaction of the wave with the projection of the object along the direction of propagation. This physical assumption is called the "pure projection approximation". With this assumption, very fine resolution 3D reconstructions can be obtained [175]. However, the depth of focus shrinks proportionally to the square of the transverse resolution (for a fixed wavelength). Hence, only minuscule objects can be imaged to

nanometer scales in the regime where the pure projection approximation holds. Nevertheless, the pure projection approximation is a computationally crucial assumption for two reasons:

1. The associated forward model is computationally efficient to evaluate.
2. The 3D reconstruction can be achieved in two distinct steps: multiple 2D phase retrieval problems are solved from different angles, and then the 2D reconstructions are merged with a tomography algorithm.

The goal of chapter 3 is to perform 3D reconstruction of objects that do not fit in the depth of focus. In this case, the *multi-slice* forward model must be used to carry out the computation of the propagation of the X-ray wave through the object [96]. In this regime, the phase retrieval problem and the tomography problem must be solved jointly, and the forward model becomes exorbitant to evaluate: the simple fact of computing the forward model for all diffractive pattern requires hundreds of core hours. Chapter 3 presents a massively parallel algorithm to perform this reconstruction and demonstrates with simulated experiments how to conduct better physical experiments outside the depth of focus.

1.2 Adversarial path planning

Another fruitful source of problems between differential equations and optimization is optimal control, where one seeks a control law which achieves some optimality criterion. Path planning is an example of an optimal control problem where the task is to find a trajectory from a specified source to a target, and the

optimality condition is based on minimizing some undesirable quantity: total time, distance, fuel or threat exposure. A particular application of path planning problems is surveillance evasion where, in the simplest scenario, an evader is choosing a path to minimize its exposure to an observer whose surveillance plan is fixed and fully known to the evader in advance.

Chapter 4 explores a surveillance evasion application where the evader only knows a finite set of possible surveillance plans but has no way of checking which of these plans is actually in place. Similarly, we assume that the observer may not observe or make use of the change of position of the evader in real time (as would be the case in satellite surveillance, for example). The evader is thus forced to design an *adversarial* plan in advance: one that anticipates the strategy of the observer. We discuss two versions of the problem. In the first one, a completely risk-averse evader seeks a trajectory minimizing his worst-case cumulative observability. In the second, the evader is concerned with minimizing the average-case cumulative observability, where this average is taken over the observer's and the evader's *mixed* strategies. That is, instead of committing to a single observation plan for the observer or trajectory for the evader, the strategies are probability distributions over observation plans or trajectories.

Although we assume neither the observer nor the evader gets to observe the strategy of their opponent, it is useful to consider the problem posed if one of the players gets to respond to the other's strategy. That is, one of the players (say, the observer) "goes first" (that is, declares his probabilistic mixed strategy) and that the other player (the evader) gets to observe that strategy and responds. As a rational agent, the evader will choose the optimal response to the observer's plan. The problem of computing the optimal trajectory in response to a fixed

strategy of the observer is called the best response problem:

Problem (Evader's best response)**.** *For a given mixed strategy of the observer, find the trajectory that minimizes the observability of the evader.*

The best response strategy can be computed by finding the viscosity solution of a Hamilton–Jacobi–Bellman PDE [25]. If the observer is obligated to play first, then he would anticipate that the evader always selects the best response. Thus, the optimal choice for the observer must be to find the strategy that maximizes the observability of the best response to that strategy. This version of the problem is called the max-min problem:

Problem (Max-min)**.** *Find the observer's strategy that maximizes the observability of the best response of the evader.*

This strategy can be computed by solving an optimization problem which involves computing best responses as a subproblem. If instead, the evader is obligated to play first, then we get the min-max problem:

Problem (Min-max)**.** *Find the evader's strategy that minimizes the observability of the best response of the observer.*

If a pair of strategies solves both the max-min problem and the min-max problem, then neither player has an incentive to deviate from these strategies since any unilateral change from one of the player will make them worse off. In this sense, the question of who plays first is irrelevant. Such a situation is called a Nash equilibrium of the game. The goal of chapter 4 is to characterize and compute these equilibria. The algorithm presented in chapter 4 is designed by borrowing ideas from convex optimization, multi-objective optimization, and optimal control.

1.3 Krylov methods for differential operators

Chapters 2 to 4 use optimization and the solution of differential equations together to solve a problem. Chapter 5, rather, uses optimization and numerical linear algebra to solve differential equations. The solution of linear differential equations is typically a two-step process:

1. The differential equation is turned into a finite dimensional linear system by a process of discretization. Among the common discretization schemes are finite difference and finite element methods, which can handle any geometry, are low accuracy and produce sparse matrices, and spectral methods that only apply to some geometries and typically produce dense matrices but are highly accurate.
2. The linear system is solved using either a direct or an iterative method.

While discretization is often necessary to obtain an algorithm that finishes in finite time, it is inconvenient for several reasons:

1. The solution of the discretization may be a poor approximation to the true solution. This is often resolved by an adaptive discretization: that is, the linear system is solved, then some criterion is used to determine whether the solution is sufficiently accurate. If it is not, then the size of the discretized system is increased and the linear system is solved again.
2. The discretization might destroy some of the computationally attractive properties of the differential equation. This is particularly true for spectral discretizations, which are typically more ill-conditioned than expected and not structure-preserving.

If the discretized linear system is small or dense, a direct method is typically used to solve the discretized system. If it is large and sparse, an iterative method is often advantageous. While direct methods are algorithms that, after a finite number of operations, compute the exact solution in exact arithmetic, iterative methods compute a sequence of approximate solutions which converge to the true solution. Krylov subspace methods are iterative methods that generate a sequence of nested subspaces (called Krylov subspaces) and choose the best approximate solution in each subspace (where the definition of "best" depends on the specific Krylov method). Chapter 5 presents analogues of these Krylov methods applied directly on the differential operators that generate continuous Krylov subspaces as opposed to Krylov subspaces that depend on a fixed discretization. In this way, they circumvent the need for discretization altogether. The idea of using Krylov methods on differential operators is not novel, in fact, the first paper which uses Krylov methods on differential operators [27] appeared only fifteen years after the paper introducing the first Krylov methods [71] in 1952. However, while previous work on the subject has been mostly of theoretical interest, the methods in chapter 5 are practical spectral methods for ordinary differential equations. They are highly efficient and forego the woes of typical discretizations: they converge to the true solution without the need for refinement and automatically preserve some structural properties of the differential operators.

CHAPTER 2

A SUBSPACE PURSUIT METHOD TO INFER REFRACTIVITY IN THE MARINE ATMOSPHERIC BOUNDARY LAYER

2.1 Introduction

The marine atmospheric boundary layer (MABL) is the part of the lower troposphere in direct contact with the ocean. This contact creates a zone of particularly high inhomogeneity due to the exchanges of heat, moisture, and momentum between the atmosphere and the ocean [152]. Within the lower MABL, the index of refraction - the speed of light in the medium relative to that in a vacuum - may change rapidly with height above the ocean surface; this causes ducting, i.e., bending of EM waves to the surface. Atmospheric ducting greatly changes the behavior of EM propagation within the MABL from what is expected in a "standard" atmosphere [153]. Ducting impacts radio communication, and it is also detrimental to maritime radars: it creates radar holes where no EM wave can travel, increases sea surface clutter, and changes the maximal operating range (illustrated in fig. 2.1). It is therefore of great interest to be able to identify and characterize the presence of ducts in real time.

A variety of methods have been developed to characterize EM ducts. A few methods link refractive index profiles and weather conditions [7] to estimate the

This chapter is based on the paper "A subspace pursuit method to infer refractivity in the marine atmospheric boundary layer" by M.A. Gilles, C. Earls and D. Bindel to appear in *IEEE Transactions on Geoscience and Remote Sensing*.

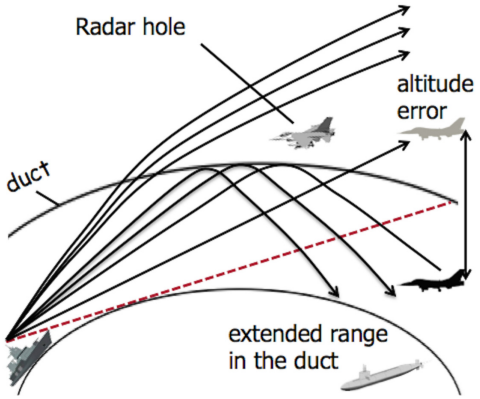


Figure 2.1: Effects of atmospheric ducting on EM waves [49].

MABL characteristics. Such estimates of meteorological conditions may come from numerical weather predictions [66, 91]. Although these numerical weather predictions are successful for long-term, general characterization, their accuracy is unsatisfactory for characterizing ducting in a local sense and in real time. Another way to estimate meteorological conditions conducive to ducting is by using radiosondes or rocket sondes [133] but those methods are costly, slow to deploy and local. Yet another way to estimate ducting conditions is by lidar [178] but this approach is sensitive to clouds, fog, and aerosols.

Other methods that rely on global positioning system (GPS) satellites have been proposed which use information about the distortion of the GPS signal to estimate refractivity profiles [101, 179]. These methods rely on the GPS being placed over the horizon with respect to the receiver, which renders this method impractical for other contexts than targets of opportunity.

In the last decade, refractivity from clutter (RFC) methods have received a lot of attention in the literature. RFC methods use a radar to estimate the refractivity profile by emitting radiation and measuring the backscattered signal from the rough ocean surface (also called clutter). RFC methods typically use either a

forward model of EM propagation or a database to "predict" the clutter under some EM condition, and compare the measurements to the prediction to infer the refractivity profiles. For a review of RFC methods, see [84].

The most popular forward model in RFC applications is the parabolic equation (PE) specialization of Maxwell's equations, briefly discussed in section 2.2. A wealth of inverse solution methods for characterizing MABL refractivity have been developed that combine the accurate and relatively fast solvers inherited by the PE with a statistical or machine-learning method. For example, [172] uses a recursive Bayesian approach, [182] uses Kalman filters, [38] uses a support vector machine, and [181] uses Bayesian Monte Carlo analysis. An example of an RFC method that learns from a database is [48]. It uses the proper orthogonal bases of collected data to form an approximate forward model to be used in an inversion aimed at characterizing the EM duct itself.

In the current paper, we are interested in a different sampling approach: the bistatic case [53, 55, 126, 128, 173, 185, 187, 188]. An example situation could involve two separate phased arrays, one transmitting and one receiving down range, with the receiver able to sample at different heights. Our method is similar to RFC methods in that it uses a forward model to predict clutter, as it also relies on a forward model of EM propagation. However, the proposed method does not involve actually solving the associated differential equation to predict the signal. Instead, we exploit a structure that is present in the partial differential equation which governs the physics: namely the approximately low-rank structure of the field within specific parts of the domain. The low-rank structure arises because only a few eigenvectors are needed to reconstruct the PDE when it is solved through separation of variables. This allows us to design an

algorithm that seeks a refractivity profile associated with eigenvectors that best fit the data. The method presented in this paper is close in spirit to the idea presented in [48, 49]. However, while they used a basis induced by the data, we use a basis induced by the forward model.

The rest of the paper is organized as follows: in section 2.2 we state some background on the problem; in section 2.3 we motivate and describe our algorithm; in section 2.4 we give details needed for a fast implementation; and in section 2.5 we present our numerical results.

2.2 Background

2.2.1 Forward problem: propagation

The physics that govern electromagnetic wave propagation are described by Maxwell's equations. Assuming a horizontal polarization and suppressing a time dependence of the form $\exp(-i\omega t)$, Maxwell's equation can be transformed into the Helmholtz equation (cf. eq. (2.1), along with useful boundary conditions eqs. (2.2) to (2.4)), by means of the following exact earth flattening transformation [134]:

$$\frac{\partial^2 f(x,z)}{\partial x^2} + \frac{\partial^2 f(x,z)}{\partial z^2} + k_0^2 n(x,z)^2 f(x,z) = 0, \tag{2.1}$$

$$\left.\frac{\partial f(x,z)}{\partial z}\right|_{z=0} = -\left(\frac{1}{2a_e} + ik_0\sqrt{\epsilon_s - 1}\right) f(x,0), \tag{2.2}$$

$$f(0, z) = F_0(z), \tag{2.3}$$

$$\lim_{r\to\infty} r\left(\frac{\partial}{\partial r} - ik_0\right) f(x, z) = 0. \tag{2.4}$$

These equations are in 2D cartesian coordinates, where x denotes the horizontal range, z denotes the vertical altitude (the direction of invariance), $r = |(x, z)|$, $k_0 = 2\pi/\lambda$ is the free-space wavenumber, λ is the wavelength, ϵ_s is the complex dielectric constant at the ocean free surface, a_e is the radius of the earth, $n(x, z)$ is the index of refraction, and f denotes the electric field in horizontal polarization. This Helmholtz equation is equipped with boundary conditions. In the case of the MABL, this is achieved by imposing continuity of the tangential field components by modeling the sea surface as a locally homogeneous dielectric and specifying a surface boundary condition [134]. This surface boundary condition is implemented via the Leontovich surface impedance condition, which for horizontal polarization is expressed as eq. (2.2). Equation (2.3) is the boundary at $x = 0$ and represents the source, i.e. the transmitter antenna. The domain is semi-infinite in both the x- and z- direction, for these boundaries radiating boundary conditions of the form of eq. (2.4) are appropriate [134].

In the particular case of the MABL, the index of refraction is often approximated to be horizontally constant [38, 181]. This assumption seems to be approximately valid for open-sea for a small region (less than 100 km) [87] but may not hold within coastal regions. In the argument that follows, we will make this assumption, and thus fix $n(x, z) := n(z)$. In section 2.5, we relax this assumption. That is, we allow for some horizontal change in the refractivity and attempt to characterize the mean refractive index, where the mean is taken over the downrange distance.

2.2.2 Index of refraction

The literature suggests that in the MABL the refraction can be well approximated by employing a modified refractivity $M(z)$, defined as

$$n = M \cdot 10^{-6} - z/a_e + 1, \tag{2.5}$$

where $a_e = 6370$ km is the radius of the earth. $M(z)$ is modeled as a tri-linear function represented by four coefficients [54]: z_b is the height of the base of the duct in meters, t_h is the thickness of the duct in meters, M_d is the M-deficit in M-units, and s_1 is the slope of the lowest linear portion in M-unit/meter. This parameterization is typically used to represent surface-base ducts [53] for which the height of the ducts is a few tens of meters. Figure 2.2 displays graphically the parametrization of M, and eq. (2.6) shows the analytical form. The slope 0.118 M-unit/m of the upper part of the refractivity profile is consistent with the mean over the whole of the United States and is not a sensitive parameter in the inversion [53].

$$M(z) = \begin{cases} M_0 + s_1 z, & z \leq z_b, \\ M(z_b) - \frac{M_d}{t_h}(z - z_b), & z_b < z \leq z_b + t_h, \\ M(z_b + t_h) + 0.118z, & z_b + t_h < z. \end{cases} \tag{2.6}$$

For the rest of the paper, we will refer to the parametrization of n, through M as $\gamma = (s_1, z_b, t_h, M_d)$. We note that M_0, the modified refractivity at the mean free surface of the ocean, could be included in the parametrization, but propagation of EM waves have been found to be insensitive to this parameter [172]; therefore following the example of the authors in [172], we fix it to a typical value of

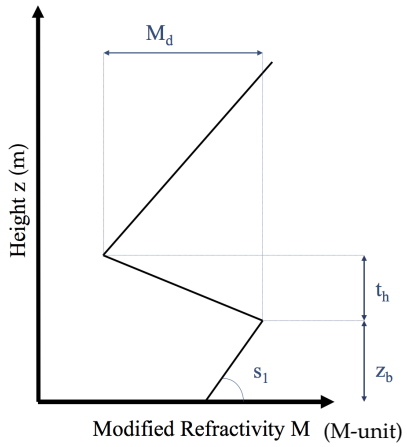


Figure 2.2: Example of a modified refractivity profile of a surface based duct. The modified refractivity profile is modeled as a tri-linear function represented by four coefficients: z_b is the height of the base of the duct in meters, t_h is the thickness of the duct in meters, M_d is the M-deficit in M-units, and s_1 is the slope of the lowest linear portion in M-unit/meter.

$M_0 = 340$. We note that the method described in section 2.3 does not rely on this parametrization and any other parametrization could be used.

2.2.3 SSFPE

The most used method for solving the PDE in eq. (2.1) together with boundary conditions of the form of eq. (2.2), eq. (2.3), eq. (2.4) is to use the split step Fourier transform for the parabolic equation method (SSFPE) [37, 121, 134, 153]. This method relies on a parabolic equation approximation of the Helmholtz equation. It is accurate, stable, relatively fast, and fairly easy to implement. Our surrogate field data is obtained by this method. In particular, our code is based on the software PETOOL, described in [121].

2.2.4 Modal solution

For simple boundary conditions and geometries, the Helmholtz equation can be solved exactly by separation of variables [81]; that is,

$$f^{\gamma}(x,z) = \sum_{m=1}^{\infty} \Phi_m^{\gamma}(x)\Psi_m^{\gamma}(z) \,,$$

where the eigenpair $(\Psi_m^{\gamma}(z), k_{rm}^{\gamma})$[1] are solutions to a Sturm-Liouville (SL) eigenvalue problem:

$$\frac{d^2\Psi_m^{\gamma}(z)}{dz^2} + \left[k_0^2 n(z)^2 - (k_{rm}^{\gamma})^2\right]\Psi_m^{\gamma}(z) = 0$$

together with the associated boundary conditions. The functions $\Psi_m^{\gamma}(z)$ are called eigenfunctions or modes, and the scalars $(k_m^{\gamma})^2$ are the associated eigenvalues (k_m^{γ} are also called associated wavenumbers). Throughout, we assume that $\Psi_m^{\gamma}(z)$ are normalized so that $\|\Psi_m^{\gamma}(z)\|_{L_2} = 1$.

For example, in the case where the source is modeled through a boundary condition at $x = 0$ (as a point source at height z_s) and the boundary condition is homogeneous Dirichlet at $z = 0$ along with homogeneous Neumann at $z = D$, the electric field solution is [81]:

$$f^{\gamma}(x,z) = \frac{i}{4}\sum_{m=1}^{\infty} \Psi_m^{\gamma}(z_s)\Psi_m^{\gamma}(z)\exp\left(ik_m^{\gamma}z\right) \,.$$

When we consider an infinite domain and more complicated boundary conditions, such as eq. (2.2) and eq. (2.4), separation of variables does not provide an exact solution. In this case, we use contour integration to obtain a solution involving a linear combination of modes from the discrete part of the spectrum and an integral term from the continuous spectrum. In practice, the integral

[1] We use γ superscripts to emphasize quantities that depend on the refractivity profile parametrized by γ.

term can be neglected if we are sufficiently far from the source [81]. The modes can be further divided into two categories:

1. Leaky modes which are not observed in range.
2. Trapped modes which propagate in range.

As noted in [81] for most long-range propagation, only the trapped modes whose wavenumber is within a certain interval of interest are important. In our case, the interval of interest contains the admissible speeds of propagation of the modes. These admissible speeds of the propagating modes are bounded by the minimum and maximum speed induced by the refractivity in the domain. In the ducting case, we are concerned primarily with the energy emitted, propagated, and received in the MABL. Therefore, by restricting our domain of dependence to the MABL, we get heuristics bounds on the set of relevant eigenvalues. Formally: we say that a solution Ψ_m^γ of the SL eigenvalue problem is one of K propagating modes if $\mathrm{Im}(k_m^\gamma) = 0$ and

$$\mathrm{Re}(k_m^\gamma) \in \left[\min_z \{k_0 n(z)\}, \max_z \{k_0 n(z)\} \right] ,$$

where the maximum and minimum are taken over a domain of dependence: $0 < z < z_{max}$. In our case, we define z_{max} to be the maximal height at which we consider non-standard refractivity. In our numerical experiments in section 2.5, we take $z_{max} = 60$ m.

Formally, we have for x large (on the order of 50 km) and $z < z_{max}$:

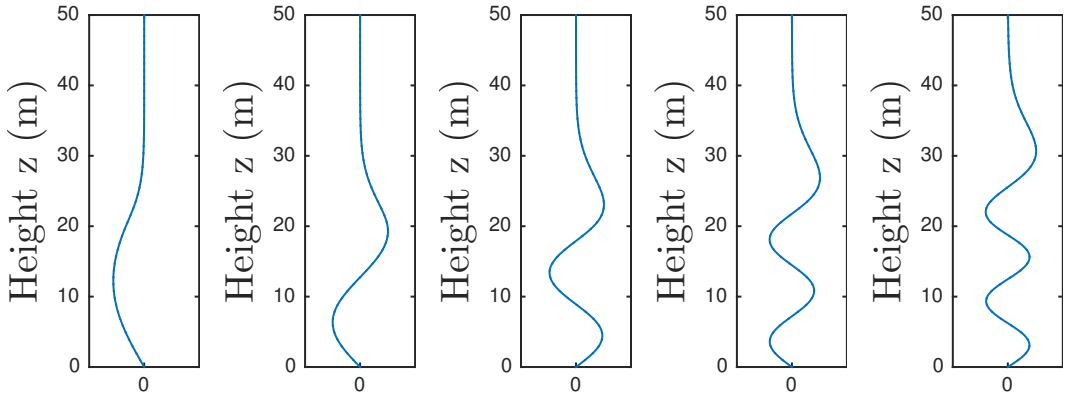
$$f_K^\gamma(x, z) \approx \sum_{m=1}^{K} a_m^\gamma \Psi_m^\gamma(z) \exp\left(i k_m^\gamma x\right) , \tag{2.7}$$

where $\Psi_m^\gamma(z)$ is a propagating mode and $a_m^\gamma = \int \bar{\Psi}_m^\gamma(z) F_0(z) dz$. In the case where we use the boundary conditions in eq. (2.2) & eq. (2.4) along with a tri-linear refractivity profile parametrized with $\gamma = (0.118, 5, 40, 30)$, the first five propagating modes are shown in fig. 2.3. Figure 2.4 shows the approximation of the field with different numbers of modes used, and the field obtained by using the SSFPE solution. Figure 2.5 shows the norm difference of $f_K(x, z)$ and $f(x, z)$ at $x = 50$ km and for $z \in [0; 30]$. We observe that after 4 modes, the field is well reconstructed.

In the rest of the paper, we will assume that the observations are collected at a fixed range, and at multiple fixed altitudes: that is we fix $x = x_{\text{obs}} \in \mathbb{R}$ and $z = z_{\text{obs}} \in \mathbb{R}^{\nu_{\text{obs}}}$. We denote

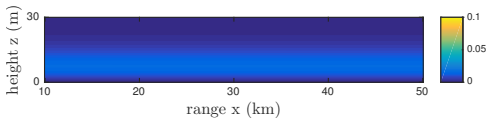
$$F(\gamma) = \begin{bmatrix} f_K^\gamma(x_{\text{obs}}, z_{\text{obs},1}) \\ \vdots \\ f_K^\gamma(x_{\text{obs}}, z_{\text{obs},\nu_{\text{obs}}}) \end{bmatrix} \in \mathbb{R}^{\nu_{\text{obs}}}, \tag{2.8}$$

where γ is the parametrization of $n(z)$.

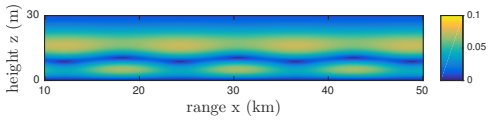


(a) mode 1 (b) mode 2 (c) mode 3 (d) mode 4 (e) mode 5

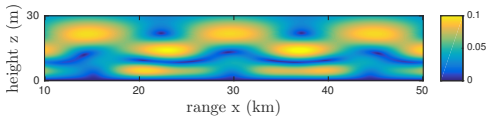
Figure 2.3: Plots of the first 5 propagating vertical modes induced by a particular refraction index parametrized by $\gamma = (0.118, 5, 40, 30)$.



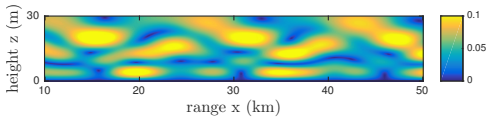
(a) 1 mode used



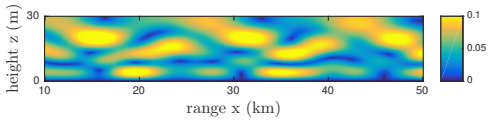
(b) 2 modes used



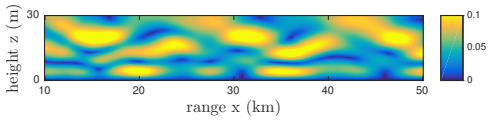
(c) 3 modes used



(d) 4 modes used



(e) 5 modes used



(f) SSFPE solution

Figure 2.4: Low order approximation of the field: $f_K(x, z)$ for increasing numbers of retained modes ((a) - (e)) and also the true field computed by SSFPE (f)

2.3 Inverse problem: characterizing refractivity

Our problem can be described in the following way: given observations of the EM response within the MABL in the form of observed data at a fixed range, x_{obs}, and different heights given by the vector z_{obs}, identify the prevailing vertical profile of the index of refraction $n(z)$. The BVP described in section 2.2 provides a forward model, which given an index of refraction, $n(z)$, lets us com-

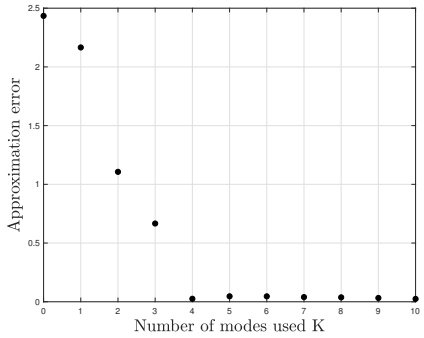


Figure 2.5: Plot of error: $\|f_K(x_{\text{obs}}, z_{\text{obs}}) - f(x_{\text{obs}}, z_{\text{obs}})\|_2$ as a function of the number of modes retained K, where $x_{\text{obs}} = 50$ km .

pute the field at any points in the domain. We would like to solve the inverse problem: find the refractivity profile given some observations of the field. A natural approach is to seek

$$\gamma^{\text{inv}} = \arg\min_{\gamma} \|F(\gamma) - F_{\text{obs}}\|, \tag{2.9}$$

where γ is a parametrization of the refractivity profile, and $F(\gamma)$ is the predicted observation under the forward model, e.g., as computed by an SSFP solver.

2.3.1 Analysis of the inverse problem properties

The objective function in eq. (2.9) has thousands of local minima, which makes global minimization very difficult. For demonstration purposes, fig. 2.6 shows a plot of a two-dimensional cut of the objective function eq. (2.9), with $F_{\text{obs}} = F(\hat{\gamma})$ for a fixed refractivity profile parameterization $\hat{\gamma} = (0.118, 5, 40, 30)$.

Although the complex function behavior is a property of the solution, and not of the modal approximation, it is illuminating to reason about this behavior in terms of the modal approximation. Using the modal approximation

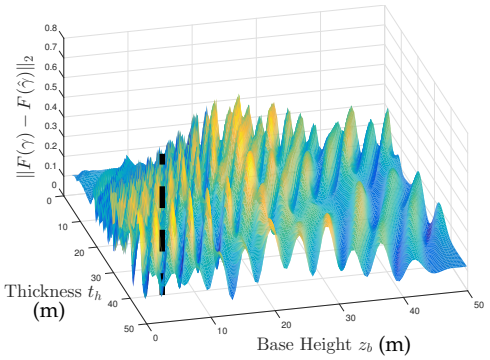


Figure 2.6: The 2-dimensional slice of the objective function eq. (2.9). The black dotted line indicates the value of $\gamma = \hat{\gamma}$, which by definition of the function is the global minimum.

in eq. (2.8), we see that in the region of interest:

$$F(\gamma) \approx \begin{bmatrix} f_K^\gamma(x_{\text{obs}}, z_{\text{obs},1}) \\ \vdots \\ f_K^\gamma(x_{\text{obs}}, z_{\text{obs},v_{\text{obs}}}) \end{bmatrix} = \begin{bmatrix} \sum_{m=1}^{K} \Psi_m^\gamma(z_{\text{obs},1}) a_m \exp\left(ik_m^\gamma x_{\text{obs}}\right) \\ \vdots \\ \sum_{m=1}^{K} \Psi_K^\gamma(z_{\text{obs},v_{\text{obs}}}) a_K \exp\left(ik_K^\gamma x_{\text{obs}}\right) \end{bmatrix} =: U(\gamma)c(\gamma),$$

where

$$U(\gamma) = \begin{bmatrix} \Psi_1^\gamma(z_{\text{obs},1}) & \dots & \Psi_K^\gamma(z_{\text{obs},1}) \\ \vdots & \ddots & \vdots \\ \Psi_1^\gamma(z_{\text{obs},v_{\text{obs}}}) & \dots & \Psi_K^\gamma(z_{\text{obs},v_{\text{obs}}}) \end{bmatrix}, \quad c(\gamma) = \begin{bmatrix} a_1^\gamma \exp\left(ik_1^\gamma x_{\text{obs}}\right) \\ \vdots \\ a_{v_{\text{obs}}}^\gamma \exp\left(ik_{v_{\text{obs}}}^\gamma x_{\text{obs}}\right) \end{bmatrix}.$$

The $U(\gamma)$ matrix spans a basis for a subspace in which the approximation is expressed and the vector $c(\gamma)$ represents the coordinates of the approximation within that subspace. Armed with this representation of the forward model, we would like to determine which of the two components causes the highly oscillatory behavior observed in fig. 2.6. Figure 2.7 shows how the subspace changes as we vary the refractivity profile γ from a reference value of $\hat{\gamma} = (0.118, 5, 20, 40)$ by plotting the largest principal angle [58] between $U(\gamma)$ and $U(\hat{\gamma})$. Figure 2.8 shows how the coordinate part of the function evolves as we change γ in the discrete 2-norm, i.e. it is a plot of:

$$\text{coor}(\gamma) = \|c(\gamma) - c(\hat{\gamma})\|_2^2.$$

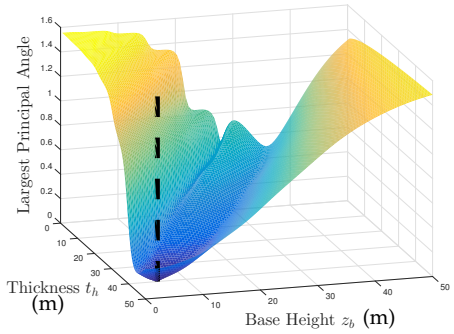


Figure 2.7: Principal angle between two subspaces: one constant, and the other induced by different refractivity profiles. The dotted line indicates the value where $\gamma = \hat{\gamma}$.

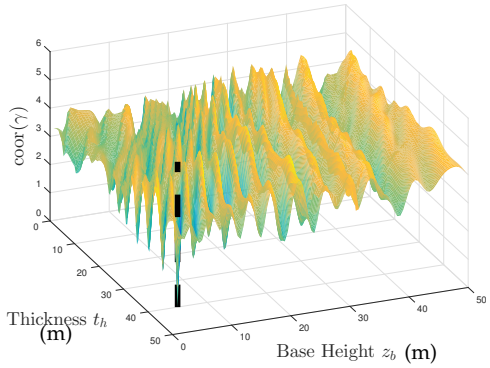


Figure 2.8: Difference of norm of "coordinates" induced by different refractivity profiles. The dotted line indicates the value where $\gamma = \hat{\gamma}$.

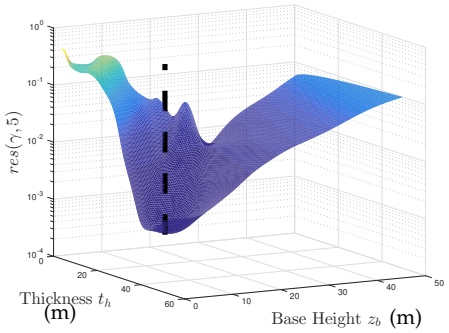


Figure 2.9: Objective function of the proposed method plotted in a semi-log scale, defined in equation eq. (2.13). The dotted line indicates the value of the true refractivity profile.

It is clear from figs. 2.7 and 2.8 that the basis $U(\gamma)$ changes smoothly, whereas the coordinate $c(\gamma)$ of the function causes the oscillatory behavior of the function

in eq. (2.9). These oscillations are explained by the terms $\exp(k_m^\gamma i x_{\text{obs}})$. Indeed, x_{obs} is typically large (here, $x_{\text{obs}} = 5 \cdot 10^4$m); hence any small change in eigenvalue k_m^γ caused by a change in γ are heavily amplified by the multiplication by x_{obs}, which causes wild oscillations of the term $\exp(k_m^\gamma i x_{\text{obs}})$.

To avoid the multimodal behavior of the objective function eq. (2.9) caused by the wild oscillations in coordinates $c(\gamma)$, we propose an alternative objective:

$$\gamma^{\text{inv}} = \arg\min_{\gamma} \|F_{\text{obs}} - \Pi_{U(\gamma)}(F_{\text{obs}})\|,$$

where $\Pi_{U(\gamma)}$ (defined precisely in the next section) is a projector onto the low-dimensional space spanned by the propagating modes for the refractivity profile parametrized by γ. In contrast to eq. (2.9), the new objective does not oscillate; see fig. 2.9.

2.3.2 Proposed inverse solution method

Our algorithm attempts to find a low dimensional subspace that best explains the measurements. To achieve this goal, we first need a measure of optimality of a subspace. Let $v \in \mathbb{C}^n$, and $\mathcal{W}$ be a K dimensional subspace of $\mathbb{C}^n$. It is natural to define the distance between the vector v and the subspace $\mathcal{W}$ as the minimum of the distance between the vector v and any vector $w \in \mathcal{W}$, as in eq. (2.10). A graphical representation of the distance between a subspace and a vector in the case where $n = 2$ and $K = 1$ is shown in fig. 2.10. Let the columns of $W \in \mathbb{C}^{n \times K}$ span $\mathcal{W}$, then the squared distance between v and $\mathcal{W}$ is defined as:

$$d^2(v, \mathcal{W}) = \min_{w \in \mathcal{W}} \|w - v\|_2^2 = \min_{\phi \in \mathbb{C}^K} \|W\phi - v\|_2^2. \tag{2.10}$$

Accordingly, we define the distance between W and a collection of m vectors

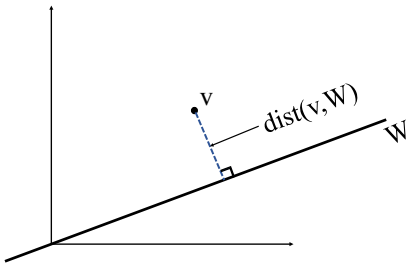


Figure 2.10: Distance between a vector $v \in \mathbb{R}^2$ and a one-dimensional subspace $W \subset \mathbb{R}^2$ (a line).

$v_i \in \mathbb{C}^n$, which are the columns of $V \in \mathbb{C}^{n \times m}$ as:

$$d^2(V, \mathcal{W}) = \sum_{i=1}^{n} d^2(v_i, \mathcal{W}) = \min_{\Phi \in C^{K \times m}} \|W\Phi - V\|_F^2. \tag{2.11}$$

Equation (2.11)) provides a way to measure the fit of some collection of measurements V into a subspace $\mathcal{W}$ spanned by the columns of W.

In our case, we have $V = F_{\text{obs}}$ (the EM observations), and $W = U(\gamma)$ (the matrix of propagating modes induced by γ sampled at the observation points). Therefore, to find the subspace which best fits the observation, we seek to perform the following minimization:

$$\min_{\gamma} d^2 \left(F_{\text{obs}}, U(\gamma) \right) . \tag{2.12}$$

According to our numerical tests, the crucial consideration to make this method successful is to characterize the dimension of the subspace properly. Indeed, it can be observed that the number of modes used, even for a fixed physical setting (i.e. frequency, boundary values, range) depends heavily on the actual refractivity profile that produces the modes. Intuitively, this can be understood in the following way: if the refractivity profile represents a very strong duct, then the trapping increases and therefore more modes are trapped

under the duct and propagate in range. In order to automatically choose the number of modes to consider, we propose two strategies:

Algorithm 1: Minimal subspace dimension

One strategy is to search over increasingly high dimensional subspaces until we find one that fits enough of the data according to some criterion (see algorithm 1). The specific criterion we use is: the objective function must be lower than some threshold value of τ, which depends on the noise level[2]. A drawback of this method is that it requires a good estimate of the noise level. We include a regularization term of Φ into the objective function. We choose the regularization parameter α by optimizing over a representative training set. We define a regularized residual function (plotted in fig. 2.9):

$$\text{res}(\gamma, k) = \min_{\Phi} \|U_k(\gamma)\Phi - F_{\text{obs}}\|_F^2 + \alpha\|\Phi\|_F^2 \ . \tag{2.13}$$

Algorithm 1 Minimal subspace dimension

Input: EM observations F_{obs}, threshold τ
$k \leftarrow 1$
$r \leftarrow \infty$
while $r > \tau$ **do**
    minimize $\text{res}(\gamma, k)$
    $r \leftarrow \min_\gamma \text{res}(\gamma, k)$
    $\gamma^{\text{inv}} \leftarrow \arg\min_\gamma \text{res}(\gamma, k)$
    $k \leftarrow k + 1$
end while
return γ^{inv}

[2]In section 2.5 we set $\tau = \eta^2 + 0.03$ where η is the noise level.

Algorithm 2: Filtering eigenvalues

Another way to determine the number of eigenvectors to include is to use an *a priori* bound on the eigenvalues to be included. As discussed in section 2.2, we can consider the propagating modes as the eigenvectors associated with eigenvalues which fall within a physically inspired interval and include only such modes. However, using such a hard threshold for included modes within an interval has some drawbacks. One is that at all places of the resulting objective function where a mode is included or excluded, the objective function is discontinuous. The other one is that our characterization of a propagating mode is physically inspired by what we define to be the domain of dependence according to z_{max}, but the true domain of dependence is infinite. Instead of a strict cut-off, we use soft thresholding by defining a filter, $t(\sigma)$, in the following way:

$$t^{\gamma}(\sigma) = \begin{cases} 1, & re(\sigma) > c_2^{\gamma}, \\ g^{\gamma}(\sigma), & c_1^{\gamma} \leq re(\sigma), \leq c_2^{\gamma} \\ 0, & re(\sigma) < c_1^{\gamma}, \end{cases} \tag{2.14}$$

where c_1^{γ} and c_2^{γ} are constants that relate to the *a priori* bounds of the eigenvalues and $g^{\gamma}(\sigma)$ is a smooth interpolation. In particular, we set

$$\begin{aligned} k_{max}^{\gamma} &= \max_z \{k_0 n(z) \mid 0 < z < z_{max}\}, \\ k_{min}^{\gamma} &= \min_z \{k_0 n(z) \mid 0 < z < z_{max}\}, \\ c_1^{\gamma} &= k_{min}^{\gamma} \\ c_2^{\gamma} &= 0.9 k_{min}^{\gamma} + 0.1 {k_{max}}^{\gamma}, \\ g^{\gamma}(\sigma) &= \hat{g}\left(\frac{\sigma - c_1^{\gamma}}{c_2^{\gamma} - c_1^{\gamma}}\right), \\ \hat{g}(\sigma) &= 6\sigma^5 - 15\sigma^4 + 10\sigma^3 \ . \end{aligned}$$

The modes which have a filtered eigenvalue of 0 are excluded from the objective function. The resulting optimization problem is:

$$\gamma^{\text{inv}} = \arg\min_{\gamma} \left\{ \hat{\Pi}(\gamma) \right\}, \tag{2.15}$$

$$\hat{\Pi}(\gamma) = \min_{\Phi} \left\{ \|U(\gamma)\Phi - F_{\text{obs}}\|_F^2 + \alpha \|t(\Sigma(\gamma))^{-1}\Phi\|_F^2 + \beta \|t(\Sigma(\gamma))\|_1 \right\} \tag{2.16}$$

where $\Sigma(\gamma)$ is the diagonal matrix of eigenvalues k_m^γ. We describe each individual term of the objective function:

- $\|U(\gamma)\Phi - F\|_F^2$

  This term is the same as defined in eq. (2.12). As discussed earlier, it models how well the data fits into the subspace spanned by the columns of $U(\gamma)$.

- $\alpha \|t(\Sigma(\gamma))^{-1}\Phi\|_F^2$

  This is the regularization term of Φ. It penalizes the contribution of Φ to modes associated with small filtered eigenvalues : that is those that we think *a priori* are less likely to be propagating. α is a regularization parameter is chosen by optimizing over a training set.

- $\beta \|t(\Sigma(\gamma))\|_1$

  The norm used in this term is the sum of the absolute values of the matrix entries. This term is the regularization term on the size of the subspace. Indeed, the objective function without this term would be naturally biased towards refractivity profiles that induce a large number of propagating modes. To give intuition as to why this happens, suppose γ_1 and γ_2 are the refractivity profiles with k propagating modes in common, but that γ_2 admits an additional $(k+1)$st mode that is not propagating for γ_1. Then, almost any data that fits γ_1 will fit γ_2 even better, even if the extra mode

only fits the measurement noise. Finally, β is a regularization parameter chosen by optimizing over a training set.

It is interesting to note that, unlike an objective function such as the one in eq. (2.9), the algorithm described in section 2.3.2 is agnostic to information about the source or the range of the transmitter. This may be useful if a good model of the source is unavailable, or if the receiver wants to be a passive listener only. The method also combines different observations from different sources or different ranges at virtually no additional computational expense which can increase accuracy.

2.4 Implementation

2.4.1 A computational shortcut

As described in section 2.2, the Sturm-Liouville eigenvalue problem that arises as a result of the separation of variables in the case we are interested in is posed on an infinite domain:

$$\frac{d^2\Psi_m(z)}{dz^2} + \left[k_0^2 n(z)^2 - k_{rm}^2\right]\Psi_m(z) = 0, \quad 0 < z < \infty, \tag{2.17}$$

$$\beta_1 \frac{\partial \Psi_m(z)}{\partial z}\bigg|_{z=0} + \beta_2 \Psi_m(0) = 0, \tag{2.18}$$

$$\lim_{z\to\infty} \hat{\alpha}_1 \frac{\partial \Psi_m(z)}{\partial z} + \hat{\alpha}_2 \Psi_m(z) = 0\,. \tag{2.19}$$

In our case, we have $\beta_1 = 1, \beta_2 = \left(1/\left(2a_e\right) + ik_0\sqrt{\epsilon_s - 1}\right)$, $\hat{\alpha}_1 = 1$, $\hat{\alpha}_2 = -ik_0$. One can solve the SL eigenvalue problem with an infinite boundary condition by solving an equivalent problem on a finite domain, but with a boundary condition that is a function of the eigenvalues, for an example of a supporting derivation, see [81]. This new SL eigenvalue problem takes the form:

$$\frac{d^2\Psi_m(z)}{dz^2} + \left[k_0^2 n(z)^2 - k_{rm}^2\right]\Psi_m(z) = 0, \quad 0 < z < D, \tag{2.20}$$

$$\beta_1 \frac{\partial \Psi_m(z)}{\partial z}\bigg|_{z=0} + \beta_2 \Psi_m(0) = 0, \tag{2.21}$$

$$\alpha_1(k_m^2)\frac{\partial \Psi_m(z)}{\partial z}\bigg|_{z=D} + \alpha_2(k_m^2)\Psi_m(D) = 0\,. \tag{2.22}$$

For the application at hand, $n(z)$ is assumed to be linear above D and therefore $\alpha_1(k_m^2)$ and $\alpha_2(k_m^2)$ can be expressed in terms of parabolic cylinder functions. One can then discretize this new SL eigenvalue problem and attempt to solve it numerically. This discretized SL eigenvalue problem gives rise to a non-linear algebraic eigenvalue problem. Nonlinear eigenvalue problems are in general expensive to solve, but since the nonlinearity is only in the boundary condition (in other words, in a single entry of the matrix), it is possible to implement fast solvers. For example, in [86] the author presents an algorithm to solve a similar problem based on Newton's method. However, in our case, this computational expense is unnecessary as we can deal with a standard linear eigenvalue problem instead. Indeed the modes that we are interested in are the so-called trapped modes. The trapped modes are those which are "trapped" physically low within the domain, and therefore their support lie below a threshold. Note

that if $\Psi_m(z)$ satisfies eq. (2.20) and eq. (2.21), and $\left.\frac{\partial \Psi_m(z)}{\partial z}\right|_{z=D} = 0$, $\Psi_m(D) = 0$, then $\Psi_m(z)$ also satisfies eq. (2.22), and therefore solves the non-linear eigenvalue problem. This implies that if we solve the linear eigenvalue problem associated with the SL problem with homogeneous Dirichlet boundary conditions at $z = D$, and the eigenvector's derivative is zero at the upper boundary, then this solution also solves the nonlinear eigenvalue problem. Otherwise, we can use this eigenpair as a first guess in Newton's method for the non-linear problem as described in [86]. In practice, we find that this step is rarely necessary provided that D is chosen high enough, and thus we only solve the Dirichlet problem to do the inference in the numerical experiments in section 2.5 in order to save the extra computational expense.

2.4.2 Computation of the modes

Implementing the algorithm involves solving the Sturm-Liouville eigenvalue problem numerically. We solve this continuous problem by discretizing using finite differences; a treatment and derivation can be found in [81]. This reduces the problem to one of computing a subset of eigenvectors and eigenvalues of a tridiagonal symmetric matrix, for which optimized solvers can be used. As pointed out in [81], one should use 5 to 10 discretization per wavelength in this type of computation. In our case, this induces a discretized system of size approximately 5000×5000. We have observed the results of the inversion to be insensitive to the discretization size chosen.

2.4.3 Inner minimization

Each function evaluation in the algorithm described above involves a minimization over Φ. However, in that inner minimization, γ is fixed. Thus the inner problem is a linear least squares problem which can be solved in closed form. Furthermore, Φ is a small matrix with a number of rows equal to the number of non-zero filtered eigenvalues (usually fewer than 10) and a number of columns equal to the number of vertical slices of sampled taken (on the order of 30). Therefore the inner minimization may be computed at the cost of a small linear solve.

2.4.4 Outer minimization

The most computationally expensive part of the evaluation of the objective function is the computation of a few eigenvectors. However, the computation of first and second derivatives of the objective function is much cheaper computationally as it only involves matrix multiplications and linear solves of small matrices. This fact, coupled with the small number of local minima of the objective function, motivated the use of derivative-based local optimization method. We use MATLAB's sequential quadratic programming method, described in [116], and perform multistart. The starting points are chosen by Latin hypercube sampling. Five starting points are used for each subspace dimension in algorithm 1, and ten starting points are used in algorithm 2.

2.5 Numerical Experiments

2.5.1 Error measures

Defining an error measure is crucial to perform the optimization over the parameters α in algorithm 1 and α and β in algorithm 2 over a training set, as well as to evaluate the performance of our method. The error measure is defined as the relative normalized ℓ_2 (RNL2) error between a trial $n(z)$ and a true $n_{\text{true}}(z)$. First we define the normalized ℓ_2 error by:

$$\text{error}_{\ell_2}\left(n(z), n_{\text{true}}(z)\right) = \frac{\int_{\xi=0}^{\xi=60}(n(\xi) - n_{\text{true}}(\xi))^2 d\xi}{\int_{\xi=0}^{\xi=60}(n_{\text{true}}(\xi))^2 d\xi}. \tag{2.23}$$

We define the RNL2 by dividing the normalized ℓ_2 error by the expected value of the normalized error of two random refractivity profiles coming from a very large representative set.

$$\text{RNL2}(n(z)) = \frac{\text{error}_{\ell_2}\left(n(z), n_{\text{true}}(z)\right)}{\mathbb{E}_{n_i(z), n_j(z)}\left[\text{error}_{\ell_2}\left(n_i(z), n_j(z)\right)\right]}. \tag{2.24}$$

The expectation is taken over the parameters for which we perform the optimization, and is approximated by averaging 10^4 trials. For example, a score of 0.1 signifies that the algorithm performs 10 times better than a random guess.

2.5.2 Experiment 1: Simulated data originating from a trilinear, horizontally constant index of refraction.

We simulate electromagnetic wave propagation by solving the partial differential equation in eq. (2.1) using the SSFPE method. We set the wavelength $\lambda = 0.1$

m, use a Gaussian antenna pattern as the source, and sample the field at a fixed range $x_{\text{obs}} = 50$ km and heights of $z_{\text{obs},j} = j$ for $j \in \{1, 2, ..., 30\}$. Therefore, each measurement $F_{\text{obs},i}$ is a vector of length 30, whose entry j corresponds to the field at range $x = x_{\text{obs}}$ and altitude $z = j$. We simulate 5 such measurements for each test case, where different measurements are obtained by varying the height (between 20 m and 30 m) and tilt (between -0.5 and 0.5 degrees off of horizontal) of the antenna (represented as a Gaussian source):

$$\text{tilt} \sim U(-0.5, 0.5), \qquad \text{height} \sim U(20, 30),$$

where $U(a, b)$ denotes the uniform distribution on $[a, b]$. This forms a matrix of measurements F_{obs} of dimension 30×5. We then contaminate the data with Gaussian white noise of standard deviation $0.3||F_{\text{obs}}||_2$. The observations are then normalized to have unit norm in order to keep the different terms of the objective function scaled relative to each other.

For the test cases, we fix $s_1 = 0.118$ M-unit/m [55], which is consistent with the mean over the whole of the United States, and has been observed to have very little variability [132]. In order to produce unbiased test cases, we generate twenty synthetic refractivity profiles by randomly sampling the parameters in the following way:

$$z_b \sim U(0, 30), \qquad M_d \sim U(0, 50), \qquad t_h \sim U(0, 30)\ .$$

These parameters are consistent with low altitude surface-based ducts. The realizations are shown in fig. 2.11. In terms of physical domain, this means that we are observing data from 0 to 30 m in altitude, and are trying to invert parameters that define non-standard refractivity profiles from 0 to 60 m. We attempt the inverse problem of identifying the refractivity profiles from the observational data. We use the algorithms described in section 2.3.2. For algorithm 1,

we set $\alpha = 10^{-4}$. For algorithm 2, we set $\alpha = 3 \times 10^{-4}$, and $\beta = 3 \times 10^{-3}$. These parameters were obtained by minimizing the RNL2 score over a separate training set and were found to be insensitive to the noise level. The result of algorithms 1 and 2 on the twenty refractivity profiles are shown in fig. 2.11.

2.5.3 Experiment 2: Simulated data originating from a trilinear, horizontally varying index of refraction.

We present results on simulated data originating from a horizontally varying index of refraction. This experiment violates the assumption that $n(x, z)$ is horizontally constant, thus one cannot expect in general to find an approximate solution of the form eq. (2.7) which induces the low-rank decomposition of the solution of eq. (2.1). It is this low-rank structure that is exploited by algorithms 1 and 2 and therefore we expect that their performance would degrade in this experiment.

Aside from the way that the refractivity profiles are generated, the setup is the same as experiment 1. The horizontal variation in the refractivity profiles is achieved by interpolating two trilinear refractivity profiles. That is, we set $n(0, z)$ to some trilinear refractivity profile, and $n(80km, z)$ to some other trilinear refractivity profile. We then pointwise linearly interpolate the two refractivity profiles (that is, not the parametrizations) between $n(0, z)$, $n(80km, z)$ to produce refractivity profiles on $n(x, z)$ for all $x \in [0, 80]$ km. As a result, the interpolated refractivity profiles are not trilinear. We define the "true" refractivity profile as the average of the refractivity profiles between the transmitter and receiver: $n_{\text{true}}(z) = 1/r \int_{x=0}^{x=r} n(x, z)dx$ where r is the range of the receiver. We set the range

of the receiver to $r = 50$ km. Note that this average will also not be trilinear. We generate twenty synthetic refractivity profiles by randomly sampling the parameters $\left(z_b^0, M_d^0, t_h^0\right)$ in the following manner:

$$z_b^0 \sim U(0, 30) \qquad t_h^0 \sim U(0, 30) \qquad M_d^0 \sim U(0, 50),$$

and the parametrization $\left(z_b^{80}, M_d^{80}, t_h^{80}\right)$ of $n(80, z)$ by sampling

$$\begin{aligned} z_b^{80} &\sim U(z_b^0 - 10, z_b^0 + 10), \\ t_h^{80} &\sim U(t_h^0 - 10, t_h^0 + 10), \\ M_d^{80} &\sim U(M_d^0 - 15, M_d^0 + 15)\ . \end{aligned}$$

This allows for a variation in the refractivity profile on the order of 20% of the maximal value of each parameter along the propagation path. We then contaminate the data with Gaussian white noise of standard deviation $0.3\|F_{\text{obs}}\|_2$. The result of algorithms 1 and 2 on these twenty synthetic test cases are shown in fig. 2.12.

2.6 Discussion

Table 2.1: Error statistics of algorithms 1 and 2 on datasets 1 and 2.

Algorithm	Dataset	Mean error	Median error	Standard deviation
1	1	0.331	0.296	0.2808
2	1	0.231	0.1982	0.129
1	2	0.441	0.370	0.2917
2	2	0.274	0.200	0.208

We observe that in all but two cases (algorithm 1 in fig. 2.11k and algorithm 2 in fig. 2.12a) the RNL2 scores are significantly below one, indicating that the

algorithm performed far better than a random guess. Qualitatively, the height and structure of the inverted ducts and true ducts are in most cases similar. Overall, algorithm 2 performs better than algorithm 1 and does not require an estimate of the noise level, thus algorithm 2 is preferable to algorithm 1. The median RNL2 score indicates that the output of the algorithm 2. is 5 times more accurate than a random guess. The addition of horizontal variation does not seem to greatly affect the quality of the inference (especially for algorithm 2, where the median RNL2 score is 0.198 and 0.200 for the horizontally constant and varying case respectively).

The two cases where the algorithms produce a RNL2 score greater than one are attributed to a failure in the minimization of the objective function and could be resolved by using more starting points in the local optimization algorithm at the cost of a higher computational expense. However, we note that the ten starting points used in algorithm 2 seem sufficient in the vast majority of cases to allow the use of local optimization algorithms. Thus, we conclude that the issue of multimodality pointed out in section 2.3.1 is largely resolved.

The timings shown in table 2.2 were performed on a single Intel i7 core processor operating at 3.60GHz. We note that the timings should be interpreted as lower bounds as the algorithms were implemented in MATLAB, and were not fully optimized.

The most important feature of this method is that it is able to overcome the highly multimodal behavior associated with the physics of EM wave propagation. This allows the use a local optimization method instead of a global optimization method which is typically used in the literature (such as genetic algorithms in, for example, [39, 49, 54, 126, 187]).

Table 2.2: Average running time on experiments 1 and 2.

Algorithm	Experiment	Avg. run time
1	1	3 min 15 sec
1	2	2 min 59 sec
2	1	5 min 12 sec
2	2	5 min 19 sec

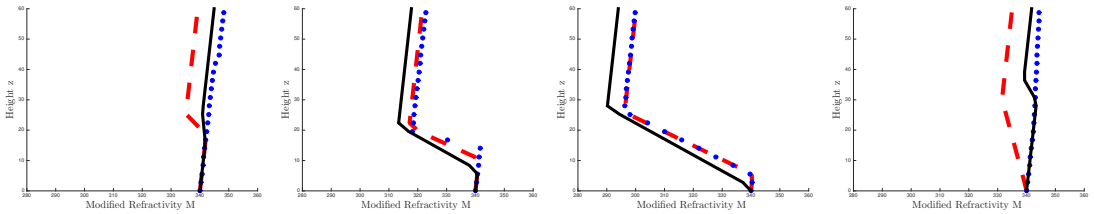
As this method is able to cheaply find an estimate of the refractivity profile using local optimization, it could also be used to warm-start a different method which would typically require a global optimization method. That is, in the first step, one could use this method to find a good first guess. Then, in the second step, a local optimization method started at that initial guess could be used to minimize a multimodal, but perhaps more accurate objective function such as the ones in [39, 54, 126, 187]. This should allow for more accurate prediction while still benefiting from the lower computational cost associated with local optimization algorithms.

2.7 Conclusion

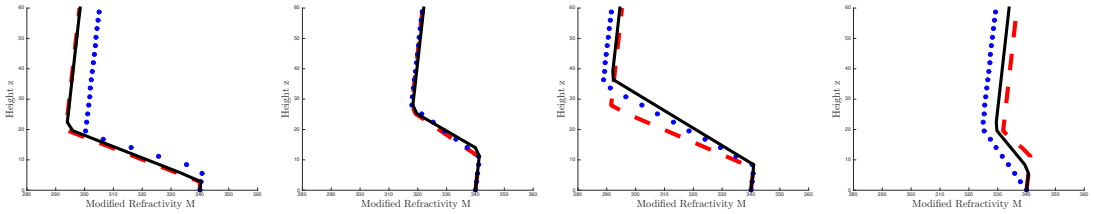
We presented a new method for characterizing the refractivity profile in the MABL which relies on the low-rank structure of the field within parts of the domain, inherited by the governing Helmholtz equation. This low-rank structure allows us to formulate the inverse problem in terms of the minimization of a new objective function. The objective function performs a projection operation of the data onto the subspace spanned by the eigenvectors that form the low-rank approximation of an electromagnetic field induced by a particular

refractivity profile. Performing an optimization on this well behaved objective function allows us to accurately solve the inverse problem in around five minutes, allowing for real-time characterization of the MABL.

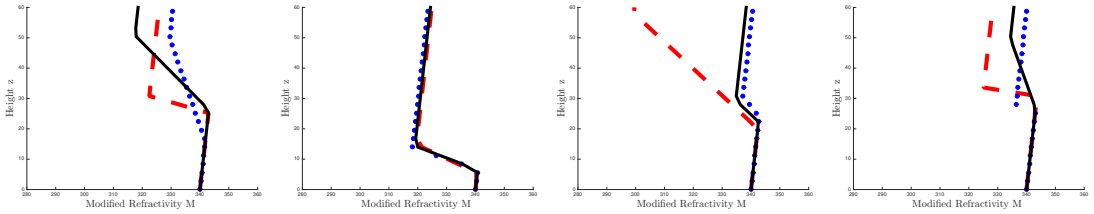
We conducted two numerical experiments to demonstrate the efficacy of the method. The first experiment was conducted on noisy simulated data computed with horizontally constant refractivity profiles, and the second experiment was performed on noisy simulated data computed with horizontally varying refractivity profiles. We observed that in both setups, the method is able to infer the refractivity profile using local optimization algorithms with few starting points thanks to the small number of local minima of the objective function.



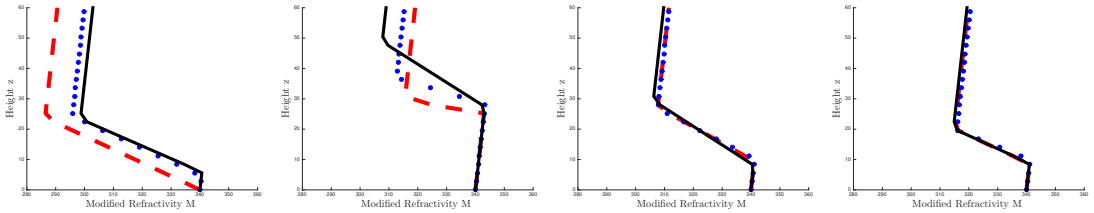
(a) RNL2 = (0.28; 0.14) (b) RNL2 = (0.25; 0.37) (c) RNL2 = (0.38; 0.37) (d) RNL2 = (0.46; 0.15)



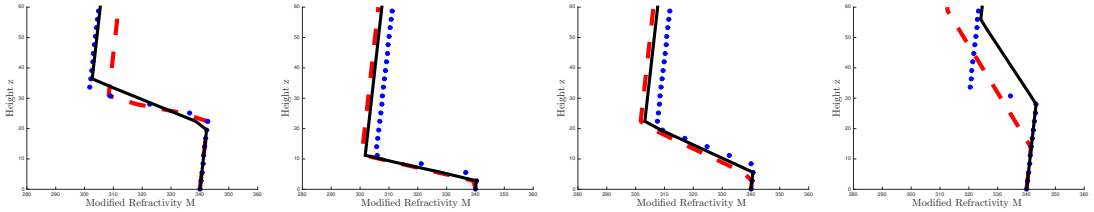
(e) RNL2 = (0.05; 0.41) (f) RNL2 = (0.07; 0.05) (g) RNL2 = (0.44; 0.21) (h) RNL2 = (0.17; 0.30)



(i) RNL2 = (0.39; 0.39) (j) RNL2 = (0.04; 0.05) (k) RNL2 = (1.12; 0.12) (l) RNL2 = (0.45; 0.17)

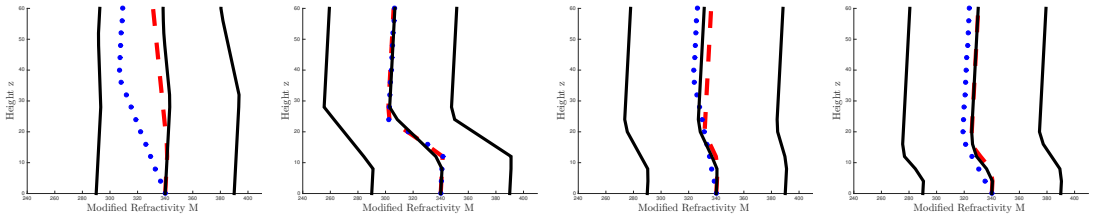


(m) RNL2 = (0.76; 0.16) (n) RNL2 = (0.58; 0.34) (o) RNL2 = (0.10; 0.11) (p) RNL2 = (0.02; 0.08)

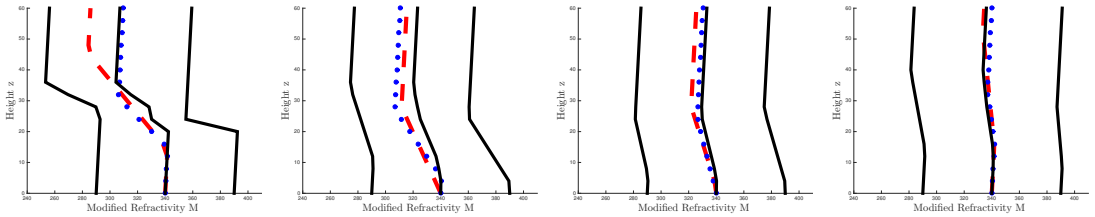


(q) RNL2 = (0.31; 0.19) (r) RNL2 = (0.07; 0.26) (s) RNL2 = (0.12; 0.25) (t) RNL2 = (0.52; 0.49)

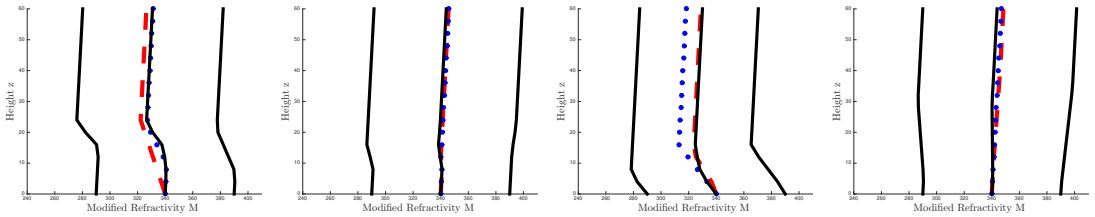
Figure 2.11: Results for algorithm 1 and 2 on experiment 1. The solid back lines are the true profile that generated the data: $n_{\text{true}}(z)$, the red dashed plot are the inverted profiles obtained using algorithm 1, and the dotted blue lines are the inverted profile obtained using algorithm 2. For each case, the RNL2 score is given below the plots. The first number is the RNL2 score of algorithm 1, and the second is the score of algorithm 2.



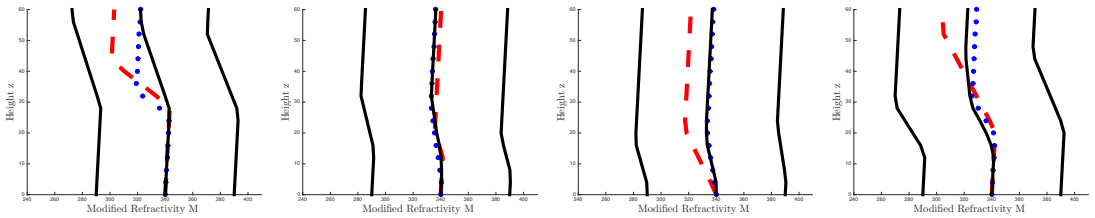
(a) RNL2 = (0.29; 1.65) (b) RNL2 = (0.12; 0.14) (c) RNL2 = (0.27; 0.23) (d) RNL2 = (0.06; 0.38)



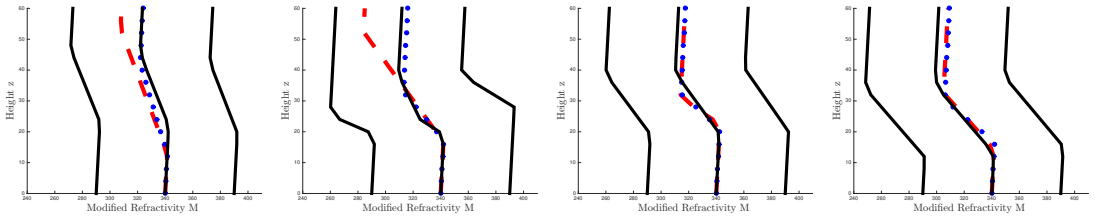
(e) RNL2 = (0.81; 0.40) (f) RNL2 = (0.52; 0.73) (g) RNL2 = (0.40; 0.18) (h) RNL2 = (0.11; 0.17)



(i) RNL2 = (0.34; 0.07) (j) RNL2 = (0.10; 0.11) (k) RNL2 = (0.10; 0.67) (l) RNL2 = (0.23; 0.17)



(m) RNL2 = (0.97; 0.52) (n) RNL2 = (0.18; 0.07) (o) RNL2 = (0.86; 0.05) (p) RNL2 = (0.55; 0.27)



(q) RNL2 = (0.55; 0.21) (r) RNL2 = (0.80; 0.19) (s) RNL2 = (0.25; 0.22) (t) RNL2 = (0.29; 0.35)

Figure 2.12: Results for algorithms 1 and 2 on experiment 2. The solid black lines on the left are $n(0, z)$, the solid black lines the right are $n(80, z)$, the solid black lines in the middle are $n_{\text{true}}(z)$, the red dashed plot are the inverted profiles obtained using algorithm 1, and the dotted blue lines are the inverted profile obtained using algorithm 2. For each case, the RNL2 score is given below the plots. The first number is the RNL2 score of algorithm 1, and the second is the score of algorithm 2.

CHAPTER 3

3D X-RAY IMAGING BEYOND THE DEPTH OF FOCUS LIMIT

3.1 Introduction

Over the entire span of the electromagnetic spectrum, x-rays offer a unique combination of nanometer wavelength to enable high spatial resolution imaging, large penetration in millimeter-scale specimens, and low values of plural and inelastic scattering to enable straightforward image interpretation and quantization [1, 3]. A variety of lens-based and lensless imaging methods have demonstrated 2D spatial resolution better than 20 nm [79, 138]. Among these, ptychography [75, 131] offers a unique combination of having a spatial resolution determined not by optics but by maximum detected scattering angle, while displaying robustness in phase retrieval for non-isolated objects [183]. In ptychography, a finite-sized x-ray probe (coherent beam spot) is used to illuminate the specimen at multiple overlapping probe positions, while recording the far-field diffraction intensity corresponding to each probe position. The resulting data redundancy allows one to recover both the object and the probe. One can even reconstruct multiple probe function modes to account for x-ray beam partial coherence [162], sample vibration [20], and continuously moving illumination [19, 31, 76, 125]. Ptychography has been used to image thin circuit layers through 300 μm of silicon at 12 nm resolution [30], sub-10 nm resolution has

This chapter is based on the paper "3D X-ray imaging of continuous objects beyond the depth of focus limit" by M.A. Gilles, Y. Nashed, M. Du, C. Jacobsen and S. Wild, *Optica* 5.9 (2018): 1078-1086.

been achieved with thinner specimens [111,147,158], and sub-wavelength resolution has been obtained using EUV light [51].

As the transverse resolution δ_r of x-ray imaging continues to be improved, a challenge lies ahead: the invalidity of the pure projection approximation. When using a circular lens, the depth of focus (DOF) of an image is given by [175]

$$\mathrm{DOF} = \frac{2}{0.61^2}\frac{\delta_r^2}{\lambda} \simeq 5.4\delta_r\frac{\delta_r}{\lambda}, \tag{3.1}$$

while $\mathrm{DOF} = 5.2\delta_r^2/\lambda$ has been found to describe the depth of focus of x-ray ptychographic images [168]. For samples with an overall dimension less than the equivalent depth of focus of the imaging approach used, each view as the object is rotated can be treated as representing a pure projection for use in a standard tomographic reconstruction algorithm. However, 3D x-ray imaging experiments that are soon within reach will involve conditions where the pure projection approximation can no longer be applied. Therefore we consider here the case of near-wavelength resolution imaging of an extended object.

Normally ptychography reconstructs a single plane, which is the exit wave leaving a 2D object or the pure projection of complex optical modulation by a 3D object. With pure projections of 3D objects, single-slice ptychographic tomography (SSPT) allows one to recover the 3D object by first applying phase-unwrapping [57] to the projection images, and then using these projections in standard tomographic reconstruction algorithms [34,62]. However, one can deal with objects located at multiple planes along the x-ray beam direction by applying the object's optical modulation at each plane and propagating the resulting wavefield to the plane of the next object slice before propagating the final exit wave to the detector plane [103]. In the reconstruction process, the probe and object functions are updated at each plane, since the ptychographic update

steps work to maximize the separability of object and probe. Using this multislice approach, images have been obtained of a few discrete, separated planes by using a single viewing direction [50,56,122,157,168] or a limited range of tilt angles [150].

One approach to 3D imaging that builds on the above work was recently demonstrated [97]. This approach, termed multislice ptychographic tomography (MSPT), involved the use of multislice ptychography to reconstruct five depth planes of a 3D object at each viewing direction (with a span over all planes sufficient to encompass the 3D object and a spacing of planes fine enough so that one can ignore to some extent Fresnel diffraction within each plane). Since the object showed primarily phase contrast, the phase projections of the sample at each angle were calculated by pixelwise addition of the phases of the five slices. This again required a phase unwrapping process to yield a pure projection image for standard tomographic reconstruction.

We demonstrate here a different approach to the reconstruction of 3D images of extended, complex objects. Rather than seeking the solution of several object planes from each viewing direction separately before combining calculated pure projections in a standard tomographic approach, we consider the totality of the 3D object in each update step. To do so, we simulate the propagation of probe function illumination waves at various probe positions and incident angles through a present guess of the 3D object so as to produce a set of assumed intensity patterns to be recorded on a far-field detector; we then adjust the guess of the object so as to minimize the difference between the actual detector plane Fourier magnitudes against our present guess of the same. Such an approach has been used with learning algorithms to guide the object updates [82,99], as

well as with the imposition of a sparsity regularizer [83]. In our case, we use a proximal alternating linearized minimization algorithm for finding the object, as will be discussed below. Our algorithmic approach exploits a parallelized implementation that can address the additional nonlinearities and computational complexity introduced with the fully propagated model, and the capabilities of high performance computing. Our approach requires no phase unwrapping because the phase shift per 3D voxel is always small, and because it is based on a forward model rather than backwards propagation of the wavefield through the object. We term our approach multislice optimized object recovery, or MOOR.

The use of multislice propagation to carry out the forward model calculation is well established. First introduced to interpret high-resolution electron microscopy data [23, 24], the multislice method has been shown [96] to recreate a wide range of x-ray optical phenomena relevant to nanoscale imaging. These include grazing incidence reflectivity and wave propagation in arbitrarily thick transmission gratings that were previously understood only by using coupled-wave theories for simple, mathematically definable structures [108,139,140,180]. Objects that have refractive index boundaries within gridded voxels can be treated by filling the voxel with a weighted sum of materials [96]. Thus, multislice propagation releases one from the limit of considering only those objects for which the Born approximation applies, or objects that are constrained to be within the effective depth of focus of the imaging scheme employed.

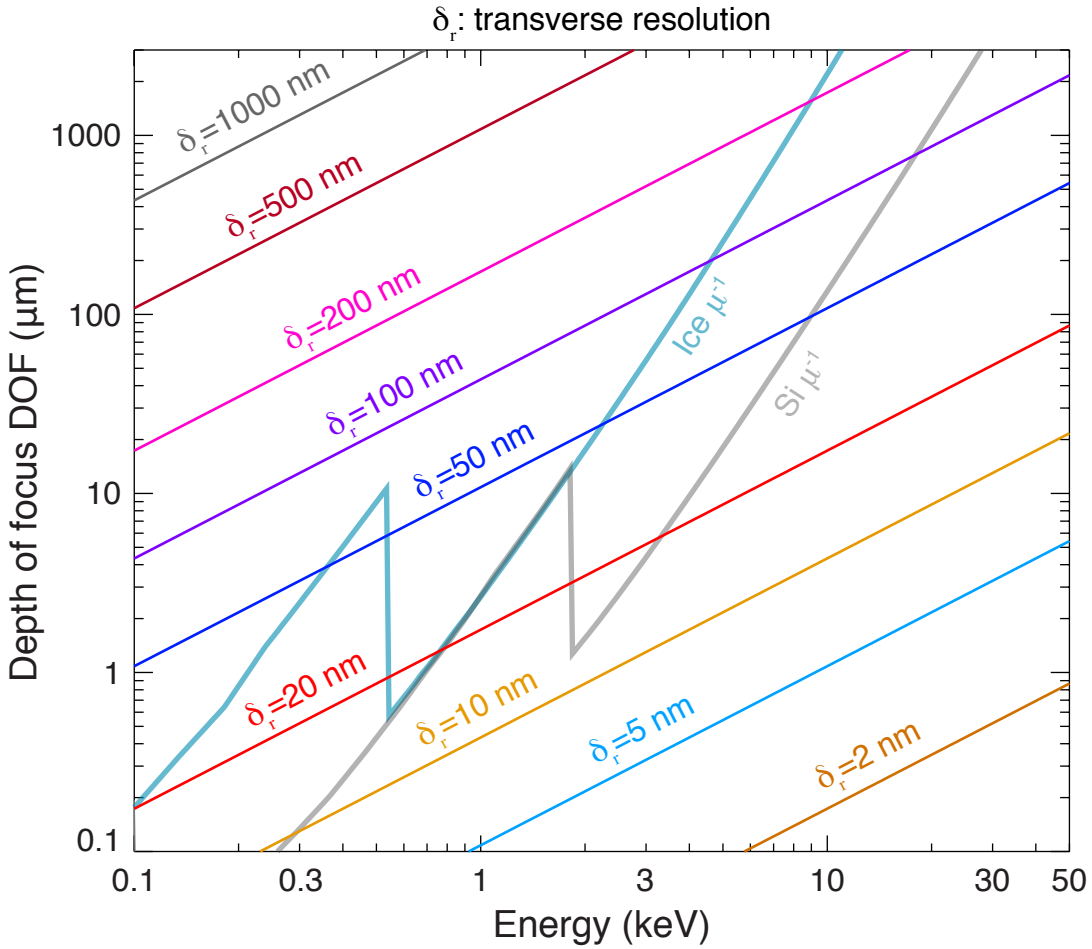


Figure 3.1: Depth of focus as a function of x-ray photon energy for a variety of transverse resolution δ_r values. Also shown is the exp[−1] penetration depth μ^{-1} for amorphous ice (for frozen hydrated biological specimens) and silicon (for microelectronics specimens) as proxies for the thickness range of x-ray imaging as a function of photon energy. As the transverse resolution δ_r in x-ray microscopy is improved to finer values, the DOF decreases with the square of the resolution improvement (eq. (3.4)), leading to a decrease in the size of a specimen that can be imaged within the projection approximation required by standard tomography.

3.2 Beyond the pure projection approximation

The Rayleigh resolution criterion [11] uses the position of the first minimum of the Airy intensity distribution as the measure of the transverse resolution δ_r,

giving

$$\delta_r = 0.61\frac{\lambda}{\mathrm{NA}}, \tag{3.2}$$

where λ is the wavelength and NA is the numerical aperture of a circular lens. We will also describe the angle of maximum scattering from the object by NA. For a circular lens, the axial intensity distribution $I(z)$ along the focal distance [100] is

$$I(z) \propto \left[\frac{\sin(u(1-b_f^2))}{u}\right]^2, \tag{3.3}$$

with $u \equiv \frac{\pi}{2}\frac{\mathrm{NA}^2 z}{\lambda}$, and with b_f as a central stop fraction. The first minimum of the longitudinal intensity distribution of eq. (3.3) occurs when $u = \pi$, giving a suggested longitudinal resolution of $2\lambda/\mathrm{NA}^2$. In fact, a more realistic criterion is to define the depth resolution δ_z as half this value, or $\delta_z = \frac{\lambda}{\mathrm{NA}^2}$, so that the DOF extends by $\pm\delta_z$ about the central focus plane, giving DOF $= 2\delta_z$. When combined with the Rayleigh resolution of eq. (3.2), the DOF can be written as

$$\mathrm{DOF} = 2\delta_z = \frac{2}{0.61^2}\frac{\delta_r^2}{\lambda} \simeq 5.4\delta_r\frac{\delta_r}{\lambda}, \tag{3.4}$$

which agrees well with experimental observations for absorption contrast imaging in a scanning transmission x-ray microscope [175] as well as with multiple-plane x-ray ptychography observations [168]. That is, as the transverse resolution approaches the x-ray wavelength, the DOF approaches the transverse resolution. Without an approach to go beyond the DOF limit, the natural combination of short wavelength (enabling high spatial resolution) and large penetration (enabling thick specimen tomography) intrinsic to imaging with x-rays cannot be fully exploited. This situation motivates the development of approaches that can work beyond the DOF limit.

Ptychography currently allows for the solution of the phase problem from a thin object or from discrete planes. However, the nature of diffraction (that it is

jointly contributed to by materials throughout the entire depth of the sample) implies that the exiting wave contains 3D information about the whole object, which inspires an iterative optimization-based solution to the object unknowns. This strategy requires a forward model to propagate the probe wave through the object in each iteration, which can be implemented by using a multislice approach. Multislice propagation decomposes the object into a number of thin layers with thickness Δz along the beam axis. For the jth layer, the incident wavefront ψ_j is modified by the well-tabulated [69] refractive index $n = 1 - \delta - i\beta$ of the material in that layer as

$$\psi'_j(x,y) \approx \psi_j(x,y) \exp\Big[\frac{2\pi\Delta z}{\lambda}(i\delta(x,y,z_j) - \beta(x,y,z_j))\Big] = M_j[\psi_j(x,y)].$$

It is then free-space propagated to the next slice. For this step, the Fresnel diffraction integral can be used and implemented as the convolution between the wavefront and a Fresnel kernel

$$h(x,y) = \exp\Big[-i\frac{\pi}{\lambda\Delta z}(x^2+y^2)\Big] \tag{3.5}$$

so that the wavefront at the $(j+1)$-th layer can be expressed as

$$\begin{aligned}\psi_{j+1}(x,y) =& \psi'_j(x,y) \circledast h(x,y)\\ =& \mathcal{F}^{-1}[\mathcal{F}(\psi'_j(x,y))H(x,y)] \qquad (3.6)\\ =& P_z[\psi'_j(x,y)], \qquad (3.7)\end{aligned}$$

where $\mathcal{F}$ denotes the Fourier transform operator and $H(x,y) = \exp\Big[-i2\pi\frac{z}{\lambda}\sqrt{1-(\lambda u_x)^2-(\lambda u_y)^2}\Big]$ is the Fourier transform of $h(x,y)$. Wavefield propagation from the exit plane to the detector is usually over a long distance L satisfying $L \gg a/\lambda$, where a is the wavefield extent. In such a scenario, the Fraunhofer approximation applies, allowing the wavefield at the detector plane to be written simply as the Fourier transform of the exit wave. If the object is divided into

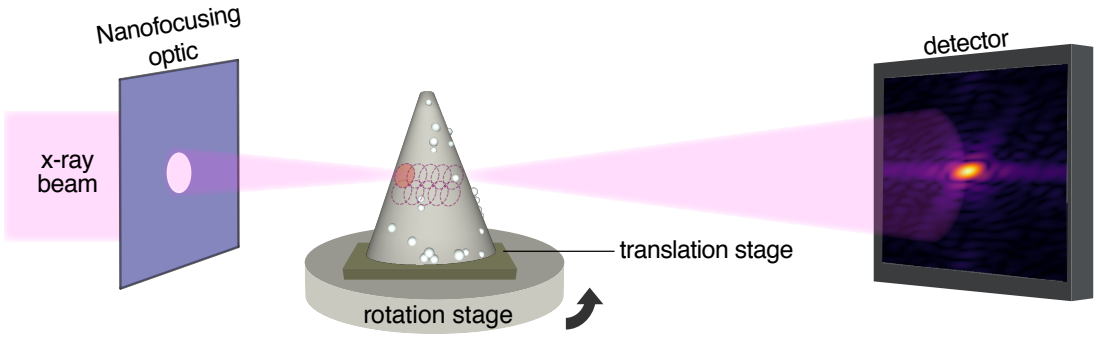


Figure 3.2: Experimental geometry used for our simulated experiment. A lens was assumed to produce a Gaussian coherent illumination probe of size 14-nm full width at half maximum (FWHM) through which the object is scanned at each object rotation angle. A far-field diffraction pattern is then captured in each scan.

s layers, the detector wavefront ψ_d will then be

$$\psi_d(x,y) = \mathcal{F}[\psi'_s(x,y)]. \tag{3.8}$$

Together, the entire forward model is

$$\psi_d(x,y) = \mathcal{F}\Big\{P_z\Big[M_s\Big[\underbrace{\cdots P_z[M_0[\psi_0(x,y)]]\cdots}_{\text{repeated nesting until } s-1}\Big]\Big]\Big\}. \tag{3.9}$$

This converts the inherently nonlinear problem of 3D diffraction into a set of nonlinear matrix equations. One can carry out multislice propagation with a layer thickness equal to the transverse pixel size or can reduce computational steps by increasing the slice thickness to be a fraction of the DOF of eq. (3.4). A good strategy is to use a voxel size that is a fraction (e.g., a tenth) of the desired spatial resolution, so as to avoid too coarse a grid, and a slice thickness that is a fraction (e.g., a tenth) of the DOF of eq. (3.4); these choices have been shown in simulations [96] to agree well with the asymptotic limit as one goes to finer voxel sizes and slice thicknesses.

3.3 Optimization methodology

We discretize the object domain with a 3D regular grid of dimension $n \times n \times n$ and discretize the probe domain with a 2D regular grid of dimension $l \times l$; the real-valued measurements are assumed to lie on a 2D regular grid of size $l \times l$. For ease of representation, we employ vector notation and denote the object variable by $\mathbf{y} \in \mathbb{C}^{n^3}$, the probe variable by $\mathbf{x} \in \mathbb{C}^{l^2}$, and each of m scan data by $\mathbf{d}_j \in \mathbb{R}_+^{l^2}$. The variable $\mathbf{y}$ is related to β and δ by

$$\beta = -\frac{\log(|\mathbf{y}|)\lambda}{2\pi\Delta z}, \qquad \delta = \frac{\arg(\mathbf{y})\,\lambda}{2\pi\Delta z}. \tag{3.10}$$

We choose to solve for $\mathbf{y}$ to avoid unnecessary nonlinear transformation during the inversion.

We let $S_{q,j} \in \mathbb{R}^{n^3 \times l^2}$ denote the linear operator that samples the qth propagation plane in the jth scan of the object. In this notation, the forward model generating scan j is

$$\left| \mathcal{F} \left\{ \left(\prod_{q=1\ldots s} C\mathrm{diag}\left(S_{q,j}\mathbf{y}\right) \right) \mathbf{x} \right\} \right| = \mathbf{d}_j, \tag{3.11}$$

where C denotes the convolution operator and the product is in the order $\prod_{q=1\ldots s} a_q = a_s a_{s-1} \cdots a_1$.

We perform the inversion by solving the constrained minimization problem

$$\min_{(\mathbf{x},\mathbf{y},\mathbf{z}) \in \mathcal{X}\times\mathcal{Y}\times\mathcal{M}} F(\mathbf{x},\mathbf{y},\mathbf{z}) + \kappa \mathrm{TV}_3(\mathbf{y}), \tag{3.12}$$

where the constraint sets are given by

$$\begin{aligned} \mathcal{X} &= \{\mathbf{x} \in \mathbb{C}^{l^2} : |\mathbf{x}| \le \nu_x\}, \\ \mathcal{Y} &= \{\mathbf{y} \in \mathbb{C}^{n^3} : |\mathbf{y} - 1| \le \nu_y\}, \text{ and} \\ \mathcal{M} &= \left\{\mathbf{z} \in \mathbb{C}^{l^2 \times m} : |\mathcal{F}\left\{\mathbf{z}_j\right\}| = \mathbf{d}_j,\ j = 1, \ldots, m\right\}. \end{aligned} \tag{3.13}$$

The constants ν_x and ν_y are used to define a compact search space and provide bounds on the magnitudes of the probe and object functions. In practice, we imagine that these values are set in a way that the minimization of eq. (3.12) is not expected to result in values of $\mathbf{x}$ and $\mathbf{y}$ that lie on the boundary of the sets $\mathcal{X}$ and $\mathcal{Y}$, respectively.

The first term of the objective function in eq. (3.12) of

$$F(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \sum_{j=1}^{m} \left\| \left(\prod_{q=1\ldots s} C \operatorname{diag}\left(S_{q,j}\mathbf{y}\right) \right) \mathbf{x} - \mathbf{z}_j \right\|_2^2 \tag{3.14}$$

is a generalization of the one used for 2D ptychography [70] and represents the data mismatch based on the forward model in eq. (3.11). The second term is composed of a scalar $\kappa > 0$ and a 3D total variation regularization term, which has proven highly effective in 3D image reconstruction [83]. This term is the 1-norm of the finite-difference approximation of the gradient of the variable $\mathbf{y}$. That is, $\mathrm{TV}_3(\mathbf{y}) = |\partial_h \mathbf{y}|_1$, where ∂_h is the finite-difference approximation of the gradient computed using the difference between neighboring voxels in each of the three spatial directions. We use the proximal alternating linearized minimization algorithm [10] to solve the minimization problem in eq. (3.12). Our algorithm is summarized in algorithm 2 and uses projections and proximal operators as in [10, 70]. The jth component of the projection onto $\mathcal{X}$ is

$$\left[\operatorname{proj}_{\mathcal{X}}(\mathbf{x})\right]_j = \begin{cases} \mathbf{x}_j, & \text{if } |\mathbf{x}_j| \le \nu_x, \\ \frac{\mathbf{x}_j}{|\mathbf{x}_j|}\nu_x, & \text{otherwise,} \end{cases} \tag{3.15}$$

and the jth block (of size l^2) of the projection onto $\mathcal{M}$ by

$$\left[\operatorname{proj}_{\mathcal{M}}(\mathbf{z})\right]_{\mathcal{J}_j} = \mathcal{F}^{-1}\left\{\hat{\mathbf{z}}\right\}, \tag{3.16}$$

where, for $w = 1, \ldots, l^2$,

$$\hat{\mathbf{z}}_w = \begin{cases} d_{j,w} \frac{\left[\mathcal{F}\left\{\mathbf{z}_{\mathcal{J}_j}\right\}\right]_w}{\left|\left[\mathcal{F}\left\{\mathbf{z}_{\mathcal{J}_j}\right\}\right]_w\right|} & \text{if } \left[\mathcal{F}\left\{\mathbf{z}_{\mathcal{J}_j}\right\}\right]_w \neq 0 \\ d_{j,w} & \text{otherwise.} \end{cases}$$

The computation of the proximal operator

$$\text{prox}_{\alpha_y}^{\tau \mathcal{Y} + \kappa \text{TV}_3}(\mathbf{y}) = \arg\min_{\hat{\mathbf{y}} \in \mathcal{Y}} \text{TV}_3(\hat{\mathbf{y}}) + \frac{\alpha_y}{2} \|\hat{\mathbf{y}} - \mathbf{y}\|^2 \tag{3.17}$$

can be performed efficiently by fast gradient projection [9]. Because of the

Algorithm 2 A proximal alternating linearized minimization algorithm for solving eq. (3.12).

PALM $\mathbf{x}^0, \mathbf{y}^0, \mathbf{z}^0, \alpha_x, \alpha_y, \alpha_z, \kappa, \nu_x, \nu_y$
for $k = 0 \cdots N$ **do**
  $\mathbf{x}^{k+1} \leftarrow \text{proj}_{\mathcal{X}}\left(\mathbf{x}^k - \frac{1}{\alpha_x} \nabla_{\mathbf{x}} F(\mathbf{x}^k, \mathbf{y}^k, \mathbf{z}^k)\right)$
  $\mathbf{y}^{k+1} \leftarrow \text{prox}_{\alpha_y}^{\tau \mathcal{Y} + \kappa \text{TV}_3(\mathbf{y})}\left(\mathbf{y}^k - \frac{1}{\alpha_y} \nabla_y F(\mathbf{x}^{k+1}, \mathbf{y}^k, \mathbf{z}^k)\right)$
  $\mathbf{z}^{k+1} \leftarrow \text{proj}_{\mathcal{M}}\left(\mathbf{z}^k - \frac{1}{\alpha_z} \nabla_{\mathbf{z}} F(\mathbf{x}^{k+1}, \mathbf{y}^{k+1}, \mathbf{z}^k)\right)$
end for
return $\mathbf{x}^N, \mathbf{y}^N, \mathbf{z}^N$

nonlinearity inherent to the forward model in eq. (3.11), updating the stepsize parameters in each iteration (e.g., by a line search procedure like in our implementation) can significantly benefit overall performance. The convergence of algorithm 2 can be further accelerated using an inertial version of the proximal alternating linearized minimization (iPALM) [127], which our implementation exploits. The evaluation of the partial gradients, discussed below, is the main computational bottleneck of algorithm 2.

3.4 Derivative computation

The gradient with respect to $\mathbf{y}$ of the multi-slice propagation operator in eq. (3.9) is:

$$\nabla_{\mathbf{y}}\left(\left(\prod_{q=1\ldots s} C\mathrm{diag}\left(S_{q,j}\mathbf{y}\right)\right)\mathbf{x}\right) = \sum_{l=1}^{s}\left(\prod_{q=l+1\ldots s} C\mathrm{diag}\left(S_{q,j}\mathbf{y}\right)\right) C\mathrm{diag}\left[\left(\prod_{q=1\ldots l-1} C\mathrm{diag}\left(S_{q,j}\mathbf{y}\right)\right)\mathbf{x}\right] S_{l,j}.$$

Thus, the partial gradient $\nabla_{\mathbf{y}} F$ is:

$$\nabla_{\mathbf{y}}\left(\sum_{j=1}^{m}\left\|\left(\prod_{q=1..s} C\mathrm{diag}\left(S_{q,j}\mathbf{y}\right)\right)\mathbf{x}-\mathbf{z}_j\right\|^2\right)$$

$$= 2\sum_{j=1}^{m}\sum_{l=1}^{s} S_{l,j}^{T}\mathrm{diag}\left[\left(\prod_{q=1\ldots l-1} C\mathrm{diag}\left(S_{q,j}\mathbf{y}\right)\right)\mathbf{x}\right]\overline{C^{*}\left(\prod_{q=s\ldots l+1}\mathrm{diag}\left(\overline{S_{q,j}\mathbf{y}}\right)C^{*}\right)\left[\left(\prod_{q=1\ldots s} C\mathrm{diag}\left(S_{q,j}\mathbf{y}\right)\right)\mathbf{x}-\mathbf{z}_j\right]}.$$

The subsampling matrices $S_{q,j} \in \mathbb{R}^{n^3\times l^2}$ contain only l^2 non-zero entries, and matrix vector-products with discretized convolution matrices $C \in \mathbb{C}^{l^2\times l^2}$ can be computed in $O\left(l^2\log(l)\right)$ thanks to the fast Fourier transform [58]; thus a naïve computation of $\nabla_{\mathbf{y}} F$ costs $O\left(ms^2l^2\log(l)\right)$ operations. However, this cost may be reduced to $O\left(msl^2\log(l)\right)$ operations (a factor of $s \approx 10^2$ less in our demonstration) at the cost of $O(msl^2)$ extra distributed memory using a dynamic programming approach outlined in algorithm 3. The computational cost of both $\nabla_{\mathbf{x}} F$ and $\nabla_{\mathbf{z}} F$ is also $O\left(msl^2\log(l)\right)$ operations.

Even with these computational savings, the computation of F and its partial gradients need to be massively parallelized across the independent refractive patterns in order to enable a large-scale solution such as that considered in our demonstration.

Algorithm 3 Algorithm to compute $\nabla_y \left(\left\| \left(\prod_{q=1..s} C \mathrm{diag}\left(S_{q,j}\mathbf{y}\right)\right)\mathbf{x} - \mathbf{z}_j \right\|^2 \right)$

$d \leftarrow \mathbf{x}$
for $l = 1 \ldots s$ **do**
  $d \leftarrow C\left(\left(S_{l,j}\mathbf{y}\right) \odot d\right)$
end for
$c_s \leftarrow d$
for $l = s - 1 \ldots 2$ **do**
  $c_l \leftarrow C\left(\left(S_{l,j}\mathbf{y}\right) \odot c_{l+1}\right)$
end for
$d \leftarrow \mathbf{x}$
$\nabla_{\mathbf{y},j} F \leftarrow 2S_{1,j}^T \left(\bar{d} \odot C^* c_1\right)$
for $l = 2 \ldots s$ **do**
  $d \leftarrow C\left(\left(S_{l,j}y\right) \odot d\right)$
  $\nabla_{\mathbf{y},j} F \leftarrow \nabla_{\mathbf{y},j} F + 2S_{l,j}^T \left(\bar{d} \odot (C^* c_l)\right)$
end for
return $\nabla_{\mathbf{y},j} F$

3.5 Demonstration

Figure 3.2 illustrates the setup used for our simulated experiment. One can increase the throughput of x-ray ptychography by using small focused beams and small-pixel-count detector arrays with high frame rate [80], so this is the configuration we selected for our simulation. We simulated a Gaussian focus spot of 5 keV or $\lambda = 0.25$ nm X rays with 14 nm FWHM probe size. This was represented with a $l \times l = 72 \times 72$ pixel array with 1 nm pixel size. We considered an object described below (200 nm in extent, sampled on a 3D grid with $n \times n \times n = 256 \times 256 \times 256$) that is large enough to go beyond the pure projection approximation at the chosen spatial resolution (1 nm voxel size, since sub-wavelength resolution imaging has been demonstrated with ptychography [51]), forcing us to account for beam propagation effects within the specimen. In other words, the "depth of focus" of diffraction effects within the object is found from eq. (3.4) to be $5.4(1\text{ nm})^2/(0.25\text{ nm}) = 22$ nm, while the probe has

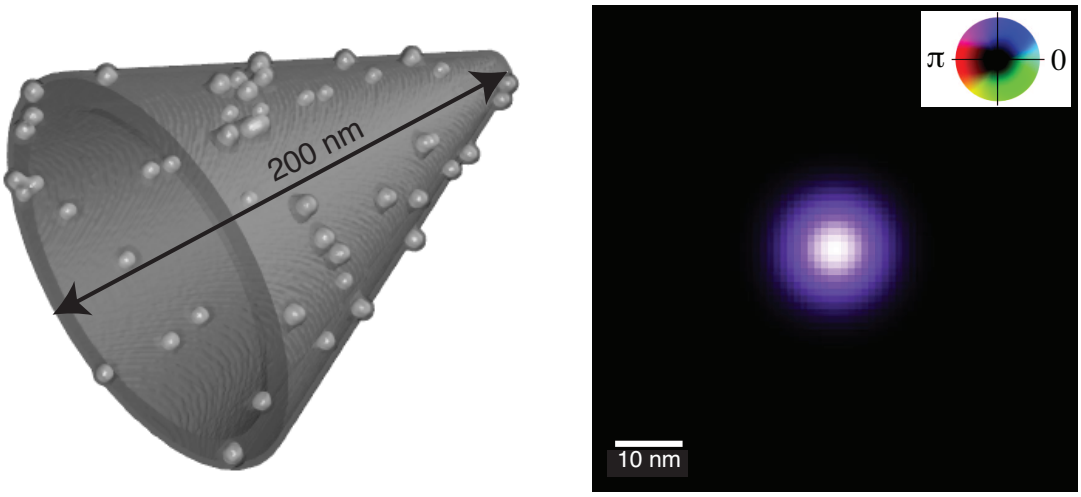


Figure 3.3: Left: an isosurface rendering of the true object with 200 nm length along the cone's axis. The simulated object was a conical glass capillary tube with embedded Ti nanospheres. Right: the as-designed Gaussian probe function with 14 nm FWHM size, as represented at the midpoint of the object region; brightness indicates the amplitude of the wave, and hue indicates the phase (see color wheel inset).

a depth of focus of 4,300 nm. With 1 nm voxel size, we were able to use nearest-neighbor sampling rather than interpolation for tomographic object rotation. Assuming Nyquist sampling of discrete Fourier transforms, a $l \times l = 72 \times 72$ pixel detector array was assumed to collect scattered signal going out to a maximum spatial frequency of $1/(2 \cdot 1\text{ nm}) = 500\ \mu\text{m}^{-1}$.

Figure fig. 3.3 shows the simulated object shape, and the incident illumination wavefront. The focus spot is considered to be at the center of the object, so that the 2D illumination function shown in fig. 3.3 is the exit wave of the probe backpropagated by half of the 3D object size, though this effect is small given the relatively large DOF of the probe compared to the object size.

The object was designed to resemble the thin capillary tubes that are widely used in laboratories as a carrier for cryo microscopy of cells in liquid suspension [90], and for applications such as electrophysiological probing or photo-

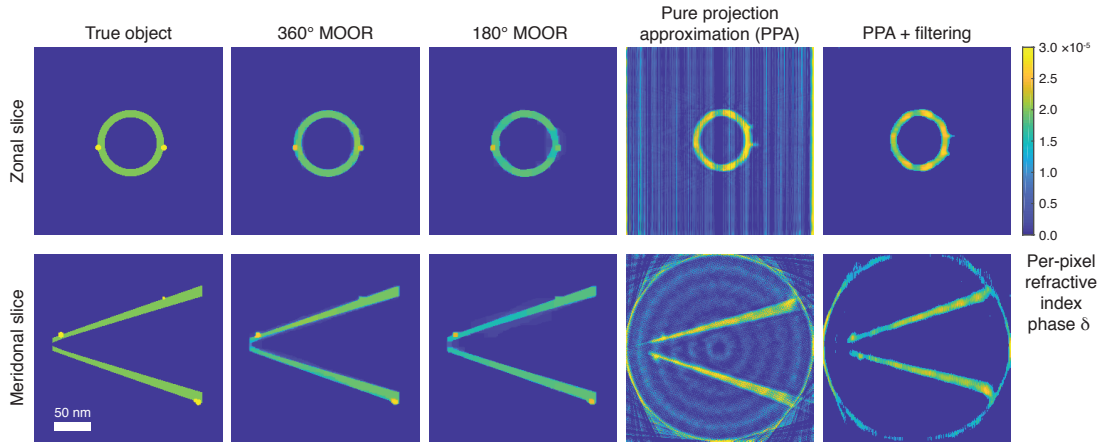


Figure 3.4: Comparisons of two cuts (zonal, top row; and meridonal, bottom row) of the reconstructed phase-shifting part δ of the x-ray refractive index $n = 1 - \delta - i\beta$ for different methods and data collections. These cuts show the reconstructed value of δ in the voxels at the selected planes. The true object is shown at left, followed by the MOOR reconstruction using 360° and then 180° rotation axes. Finally, a reconstruction from the standard pure projection approximation (PPA) to ptychographic tomography is shown. As noted in the text, the pure projection approximation does not properly reproduce the object, and it also suffers from regular artifacts due to insufficient probe overlap for the reconstruction method used [78]. For this reason, we also show a column of "PPA+filtering" images with post-processing. These figures show that 360° MOOR gives a reconstructed image that represents the true object with a high degree of fidelity.

catalysis [85]. Composite structures like these are interesting subjects for x-ray 3D imaging because of the presence of features ranging from nanospheres with diameters of tens of nanometers, to quantum dots several nanometers in size, close to the wavelength of soft x-ray probe beam. Further challenges in imaging are contributed from the capillary tube substrate, whose diameter can be orders of magnitude higher than that of the finer features, rendering the projection approximation no longer valid. The object was designed by using the open-source package *XDesign* [18], using x-ray refractive index data retrieved from a database within *Xraylib* [141]. It is composed of the tip region of a capillary tube made of Si (with $\delta = 1.98 \times 10^{-5}$ and $\beta = 1.13 \times 10^{-6}$ at 5 keV in the

complex refractive index $n = 1 - \delta - i\beta$). Nanospheres of TiO_2 ($\delta = 3.01 \times 10^{-5}$, $\beta = 3.58 \times 10^{-6}$ at 5 keV), with the diameter ranging from 8 to 20 nm, were computationally "deposited" on the outer surface. The cone-shaped capillary tip has a top diameter of 20 nm, a bottom diameter of 160 nm, and a length of 200 nm. The tube wall thickness varies linearly from 5 nm at the top to 15 nm at the bottom.

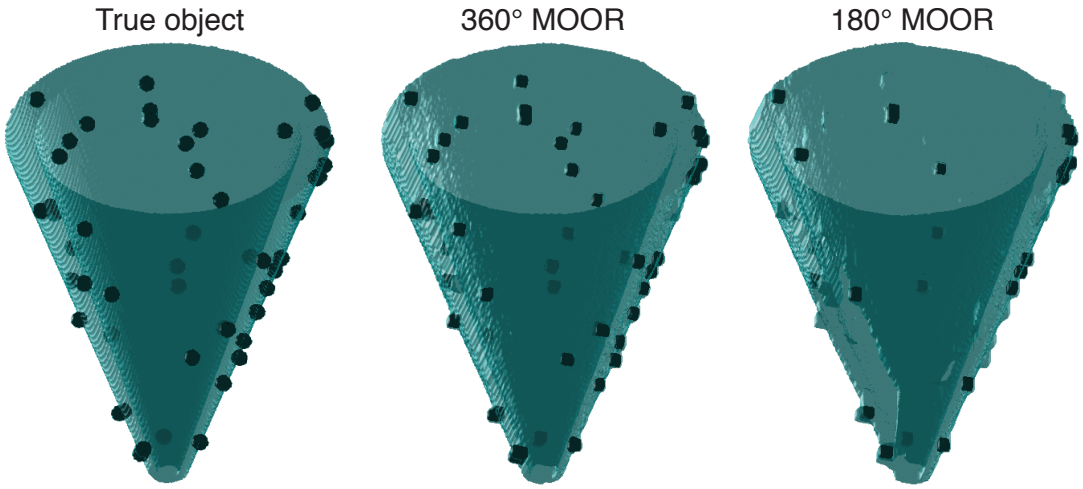


Figure 3.5: Comparisons of 3D isosurface renderings of the true object, and MOOR reconstructions using 360° and 180° object rotation. Rotation over a full 360° range gives improved results. The 3D object reconstructed using the pure projection approximation (PPA) is not shown, as its errors (fig. 3.4) do not make it possible to obtain a clean isosurface rendering.

This demonstration involved a complex object array with $n^3 = 256^3$ voxels, which implies that we are inverting for approximately 17 million complex variables. We simulated the data acquisition scans in the following way. First, data from 23^2 probe positions from a single large plane were generated. The center of each probe position on the plane was at $[12j, 12k]$ for $j, k \in 0, \ldots, 22$. We tested two rotation sampling schemes: one with 90 specimen rotations at 4° increments about a single axis over an 360° range, and another with 180 rotations at 2° increments over a 180° range. The total number of far-field diffraction patterns

recorded for both sampling schemes was thus $m = 90 \cdot 23 \cdot 23 = 47610$. We note that the 2° rotation increment is not fine enough to meet the Crowther criterion for fine angular spacing in standard tomography [26]. Finally, while our simulation did not include the effects of noise due to finite photon statistics, we note the following:

- One can arrive at a simple model for the required photon fluence to image a given feature type with a specified spatial resolution and signal-to-noise ratio [43]. In x-ray ptychography experiments, this estimate has been found to agree within 20% of what is required for imaging integrated circuit features [30] and frozen hydrated biological specimens [32] using iterative phase retrieval algorithms. This indicates that the algorithms themselves can be robust to noise, and work at the limit of the minimally required photon exposure.
- Simulation studies of the effects of photon noise in iterative algorithms for reconstructing coherent diffraction imaging data have shown that the achievable resolution is no different than conventional imaging with the same photon fluence [77]. This conclusion is consistent with other studies of the effects of photon noise on coherent diffraction imaging with X rays [142, 177].

Therefore we expect that our MOOR approach should work within the limits of the resolution that is achievable given a particular feature contrast and photon fluence.

Algorithm 2 was implemented in C and parallelized across m diffraction patterns in eq. (3.12) using the Message Passing Interface (MPI). In particular, the

reconstructed object defined by 17 million complex variables shown in figs. 3.4 and 3.5 was obtained by running the MOOR algorithm on 2,880 cores for a total of 350,000 core hours. We set the initial guess of the variables $\mathbf{y}^0 = 0$ and $\mathbf{z}_j^0 = \mathcal{F}^{-1}\left\{\mathbf{d}_j\right\}$; the magnitude of the probe variable $\mathbf{x}^0$ is initialized with $|\mathbf{x}^0| = \mathcal{F}^{-1}\left\{\frac{1}{m}\sum_{j=1}^{m}\mathbf{d}_j\right\}$ and its phase is set to a Gaussian function. The stepsize parameters α_x and α_z were fixed to $4m$ and 2, respectively, throughout the algorithms. The stepsize α_y for the object variable was initialized to 100 and updated using a line search at each iteration. These parameters were chosen to ensure the convergence of algorithm 2. The constraint variables ν_x and ν_y were both set to 1, although none of the constraints associated with these variables are active at the reconstructed image. We set the regularization parameter $\kappa = 0.01$ and ran algorithm 2 for a time budget that corresponded to approximately 600 iterations. This algorithm yielded successful reconstructions that agree well with the designed object, as shown in the images of figs. 3.4 and 3.5 and also in the quantitative values of the reconstructed phase δ as shown in the histograms of fig. 3.6.

For comparison with the (erroneous in this case) assumption that the object can be described by the pure projection approximation (PPA), we also reconstructed the simulated experimental dataset using the single slice ptychographic tomography (SSPT) approach [34, 62] discussed in the introduction. We first reconstructed a set of 2D complex-valued projections using 2,000 iterations of a GPU implementation [114] of the ePIE algorithm [105]. To align all phase projections and account for linear and constant phase terms that typically plague independent ptychographic reconstructions of the same sample, we subtracted a best-fit plane from each reconstructed phase [62]. The aligned phase and magnitude projections were then tomographically reconstructed by using 200 itera-

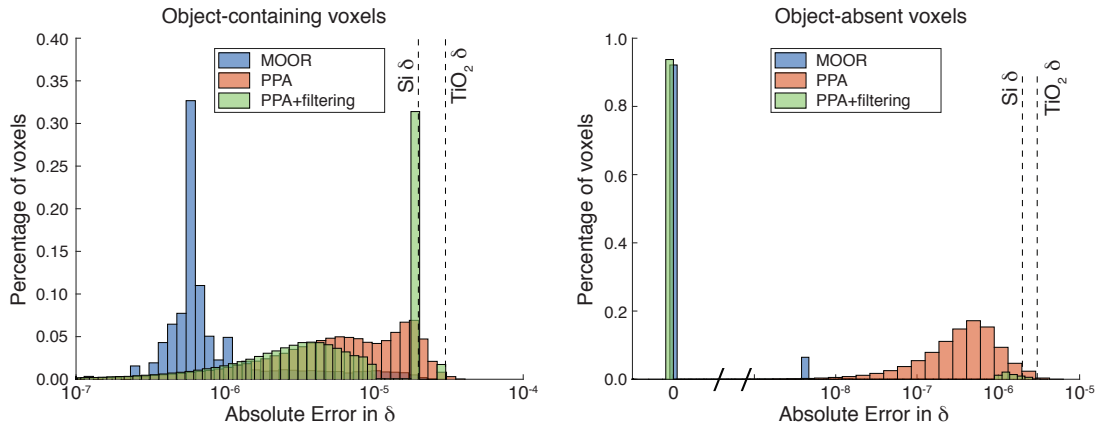


Figure 3.6: Histograms of the distribution of absolute error in the per-voxel values of the phase shifting part δ of the x-ray refractive index $n = 1 - \delta - i\beta$, for both our MOOR reconstruction approach and for the filtered pure projection approximation (PPA, and PPA+filtering) reconstructions. At left are shown the histogram for object-containing voxels, while at right are shown the histograms for object-absent voxels. Also indicated are the values of δ for Si (1.98×10^{-5} at 5 keV) and TiO_2 (3.01×10^{-5} at 5 keV). As can be seen for the object-containing voxels (left), the MOOR reconstruction approach gives results with absolute errors that are small compared to the actual values of δ, while for PPA the absolute errors are almost as large as the expected values of δ. For the object-absent voxels (right), again the errors in the MOOR reconstruction are very small compared to the expected values of δ in the object-present regions, while the PPA reconstructions have errors in the object-absent voxels that approach the true values expected in the object-present voxels.

tions of the simultaneous iterative reconstruction tomography (SIRT) method as implemented in the TomoPy package [65]. We note that the employed ptychography and tomography algorithms assume a pure projection forward model, so their poor performance shown in fig. 3.4 is expected in this case where the pure projection approximation does not apply. For example, propagation fringes at the boundaries of the sample are clearly visible, along with periodic artifacts caused by the relatively low number of projections (which can be suppressed by Fourier filtering [78]) and ePIE's inability to retrieve the correct illumination function in the circumstances of our demonstration. In order to provide a fair comparison between the pure projection approximation (PPA) and our MOOR

approach, we post-processed the result of the PPA reconstruction (we emphasize that no post-processing steps were used for the MOOR algorithm results) to give a secondary result that we have labeled as "PPA+filtering." First, we used median filtering with a window length of 10 in order to suppress the line artifacts that can be observed in the PPA column of fig. 3.4; we then thresholded the values of δ of the filtered image with a threshold of 10^{-5}.

3D ptychographic experiments typically only sample 180° around the object. This is a valid procedure under the pure projection approximation: in this setup two samples taken from symmetrically opposed locations would, in theory, produce the exact same projection image and corresponding diffraction pattern. However, situations beyond the pure projection approximation (thus requiring the multislice approach) break this symmetry, and our numerical results show that in this case a sampling around all 360° is necessary. Figure 3.4 shows the comparison of four ptychographic reconstructions with the true object: MOOR using 180° sampling, MOOR using 360° sampling, a reconstruction from a pure projection approximation (PPA) and a post-processed PPA reconstruction. In addition, fig. 3.5 shows isosurface renderings of the true object and the 360° and 180° MOOR reconstructions. Together these results show that the "hidden" side of the object is poorly resolved when using MOOR with only 180° rotation sampling.

3.6 Discussion and summary

Single-slice ptychographic tomography (SSPT) is extremely successful [34, 62, 73, 74] at obtaining high quality 3D x-ray reconstructions of objects to which the

pure projection approximation applies. For objects with larger extent, the multislice ptychographic tomography (MSPT) approach [97] has shown promise for treating the object as being represented by several planes along each viewing direction, which can be reconstructed and added to yield approximations of a pure projection. With multislice optimized object recovery (MOOR), we take the approach of allowing for multiple slice propagation through thick objects along each viewing direction; an optimization approach is then used to recover a 3D object that is consistent with all diffraction measurements, and which requires no phase unwrapping. Although we have not made an explicit comparison with the MSPT approach, the fact that one obtains an improved reconstruction with a 360 degree rotation relative to a 180 degree rotation means that a simple summation of separate reconstruction planes may not be sufficient to accurately reconstruct objects with increasing sizes beyond that to which the pure projection approximation applies.

In this first demonstration of the MOOR approach, a proximal alternating linearized minimization algorithm is used to obtain rapid convergence for the case of ptychography (where a small coherent probe is scanned across the entire specimen projection field of view at each rotation angle). This approach is consistent with the expectations of SSPT and MSPT, where the iterative rules alternate between updates of the object and the probe [105, 161]. However, our approach employs the capabilities of high-performance computing to carry out a 3D calculation with isotropic voxel size, and no need for phase unwrapping. It could also be used for other situations, such as probe overlap during rotation rather than at each viewing angle [64]. Our approach might also be useful in situations where the probe is larger than the object and hence the overlapping probe feature of ptychography is absent [155].

Although our approach is successful in recovering a computer-synthesized 3D object to which the pure projection approximation does not apply, there is room for future development. In this first demonstration, data parallelism was used so that each computing node owned and updated only a subset of the variables $\mathbf{z}_j$ (and associated scans $\mathbf{d}_j$) but also a copy of the full 3D object, which, in turn, created a synchronization point in the parallel computation for merging those copies after every update step. This allowed us to show that the approach works, but in an inefficient manner where data communication (between all the nodes with each one calculating a different probe position, and the 3D data) limited computational throughput. This limitation could be overcome by exploiting domain decomposition (i.e., parallelism in the object/reconstruction domain) for partitioning the 3D object based on the scanning pattern and the associated sparse intersections of localized illumination probes with the overall object. One could then have each node hold the probe function, a local subregion of the object, and the measured Fourier magnitudes for rapid computation with periodic synchronization happening only where the subregions overlap (we have used such an approach in standard ptychography reconstructions [114, 115]). In addition, a stochastic asynchronous version of the proximal alternating linearized minimization algorithm [29] is likely to reduce the computational cost of the algorithm as well as the synchronization cost. We will also examine the capability of our algorithm with ptychography experiments for nanoscale objects including nanofabricated devices and subcellular structures of eukaryons. Technical challenges such as probe alignment (already addressed in 2D ptychography [63, 104, 184]) could be addressed when applying the algorithm to experimental results.

CHAPTER 4

ADVERSARIAL PATH PLANNING

4.1 Introduction

Path planning is a problem of interest for many communities: traffic engineering, autonomous driving, robotics, and military. In the classical setting, the path planner is assumed to have full information about the environment and chooses a path minimizing some undesirable quantity; e.g., time-to-target, distance traveled, fuel consumption, or threat exposure. A particular type of continuous path planning problems is surveillance-evasion applications. In the simplest scenario, an evader (E) is choosing a path to minimize its exposure to an observer (O) whose surveillance plan is fixed and fully known to E in advance. This formulation is conveniently treated in the framework of *optimal control theory*, reviewed in section 4.2, with the evader's optimal policy recovered by solving a Hamilton-Jacobi-Bellman (HJB) partial differential equation (PDE). But the real focus of this paper is on path planning under uncertainty, where E knows the full list of different surveillance plans available to O but does not know which of them is currently in use.

The key assumption of our model is that neither E nor O can change their respective strategy in real time based on the opponent's discovered position or

This chapter is based on the paper "Evasive path planning under surveillance uncertainty" by M.A. Gilles and A. Vladimirsky and was submitted to *Dynamic Games and Applications* on December 26, 2018.

actions. In practical contexts (e.g., in satellite-based surveillance), this restriction might be due to either a delayed analysis of observations or due to logistical needs of committing to a strategy in advance. As in many other optimization under uncertainty situations, it is natural for E to treat this as an *adversarial* problem – either because the prior statistics on the frequency of use for specific surveillance plans are unreliable or because O might be actively adjusting these frequencies in response to E's routing choices.

In considering each potential path to its destination, E needs to evaluate the trade-offs in observability with respect to different surveillance plans. This naturally brings us to the notion of *Pareto optimality* [107] and the numerical methods developed for multi-objective optimal control problems [33, 61, 89, 113]. As we show in section 4.3, the method introduced in [89] can be used to find the deterministic optimal policy for a completely risk-averse evader (i.e., minimizing the worst-case observability). Unfortunately, the computational cost of this approach grows exponentially with the number of surveillance plans available to O. But if the goal for both players is to optimize the average-case/expected observability, we show that this can be accomplished by adopting a much more computationally affordable method from [113], despite its significant drawbacks for general multi-objective control problems. Moreover, we show that, if the evader's average-case optimal strategy is deterministic, then that same strategy is also worst-case optimal.

For the rest of the paper, we concentrate on the average-case observability formulation using a semi-infinite zero-sum game [163] between E and O, each of them searching for the best randomized/mixed strategy – an optimal probability distribution over that player's available deterministic/"pure" options.

We refer to these as "Surveillance-Evasion Games" (SEGs), although the same terminology was previously used in the 1960s and 1970s to describe a very different class of problems, where the Evader needs to escape from the Observer's surveillance zone as quickly as possible [36,93–95]. Aside from this terminological overlap, those earlier papers have little in common with our context since in them E and O operated with full information on their opponent's current state, reacted in real time, and sought optimal feedback policies recovered by solving Hamilton-Jacobi-Isaacs equations.

In classical (finite zero-sum two-player) strategic games, the Nash equilibrium is typically obtained using linear programming [120]. But the fact that E's set of pure strategies is uncountably infinite makes this approach unusable in our SEGs. Instead, we show how to compute the Nash equilibrium in section 4.4 by combining convex optimization with fast numerical methods for HJB equations. The computational cost of the resulting method scales at most linearly with the number of surveillance plans. We illustrate this approach on a large number of examples, with the details of our numerical implementation covered in section 4.5.

We note that the same ideas are also useful outside of surveillance-evasion context, whenever the path planner cannot assess the actually incurred running cost until it reaches the target. In fact, the same PDEs and semi-infinite zero-sum games can be used to model civilians' routing choices in war zones and other dangerous environments, minimizing their exposure to bomb threats.

Our modeling approach is quite general, but to simplify the exposition we will assume that the evader is moving in a two-dimensional domain with occluding/impenetrable obstacles, both the observability and E's speed are

isotropic (i.e., independent of E's chosen direction of motion), and all O's surveillance plans are stationary (i.e., the observer is choosing among possible stationary locations). This further simplifies the PDE aspect of our problem from a general HJB to stationary *Eikonal* equations, the efficient numerical methods for which are particularly well-developed in the last 25 years (e.g., [145]).

In section 4.6, we generalize the problem by considering multiple evaders. We first treat this as a two-player game between a single observer and a centralized controller of all evaders. But we also show that the resulting set of strategies is a Nash equilibrium even from the point of view of individual/selfish evaders. We conclude by discussing further extensions and limitations of our approach in section 4.7.

4.2 Continuous path planning

The case where the observer's strategy is fixed and known can be easily handled by methods of classical optimal control theory. The goal is to guide an evader (E) from its starting position $\mathbf{x}_S$ to its desired target $\mathbf{x}_T$ while minimizing the "cumulative observability" (also called "cumulative cost") along the way through its state space represented by some compact set $\Omega \subset \mathbb{R}^d$. More precisely, we will suppose that A is a compact set of control values, and $\mathcal{A}$ is the set of E's admissible controls which are measurable functions $\boldsymbol{a} : \mathbb{R} \mapsto A$. The evader's dynamics are defined by $\mathbf{y}'(t) = \boldsymbol{f}(\mathbf{y}(t), \boldsymbol{a}(t))$, with the initial state $\mathbf{y}(0) = \mathbf{x} \in \Omega$. (Even though E only cares about the optimal trajectory from $\mathbf{x}_S$, the method we use encodes optimal trajectories to $\mathbf{x}_T$ from all starting positions $\mathbf{x}$.) In all of our numerical examples, we will assume that E's state is simply its position, $\boldsymbol{f}$ is its

velocity defined on a known map Ω that excludes (impenetrable, occluding) obstacles, and E is allowed to travel along $\partial\Omega$ (including the obstacle boundaries). Suppose $T_{\boldsymbol{a}} = \min\{t \geq 0 \mid \mathbf{y}(t) = \mathbf{x}_T\}$ is the travel-time-through-Ω associated with this control. A pointwise observability function (also called cost function) $K : \Omega \times A \mapsto \mathbb{R}$ is then defined to reflect O's surveillance capabilities for different parts of the domain, taking into account all obstacles/occluders and E's current position and direction. The cumulative observability is then defined by integrating K along a trajectory corresponding to $\boldsymbol{a}(\cdot)$ with initial position $\mathbf{x}$

$$\mathcal{J}(\mathbf{x}, \boldsymbol{a}(\cdot)) = \int_0^{T_{\boldsymbol{a}}} K(\mathbf{y}(t), \boldsymbol{a}(t))\, dt, \tag{4.1}$$

which we will also denote as $\mathcal{J}(\boldsymbol{a}(\cdot))$ when $\mathbf{x}$ is clear from the context. As usual in dynamic programming, the *value function* is then defined by minimizing the cumulative observability: $u(\mathbf{x}) = \inf_{\boldsymbol{a}(\cdot)} \mathcal{J}(\mathbf{x}, \boldsymbol{a}(\cdot))$, with the infimum taken over controls leading to $\mathbf{x}_T$ without leaving Ω (i.e., $T_{\boldsymbol{a}} < \infty$ and $\mathbf{y}(t) \in \Omega, \forall t \in [0, T_{\boldsymbol{a}}]$ along the corresponding trajectory). Under suitable "small-time controllability" assumptions [6], it is easy to show that u is locally Lipschitz on Ω. If it is also smooth, a Taylor series expansion can be used to show that u satisfies a static Hamilton-Jacobi-Bellman PDE:

$$\min_{\boldsymbol{a} \in A} \{K(\mathbf{x}, \boldsymbol{a}) + \nabla u(\mathbf{x}) \cdot \boldsymbol{f}(\mathbf{x}, \boldsymbol{a})\} = 0, \quad \forall \mathbf{x} \in \Omega \setminus \{\mathbf{x}_T\}; \qquad u(\mathbf{x}_T) = 0, \tag{4.2}$$

with the special treatment at $\partial\Omega \setminus \{\mathbf{x}_T\}$ where the minimum is taken over the subset of control values A that ensure staying inside Ω.

Unfortunately, the value function u is generically non-smooth, and there usually are starting positions with multiple optimal trajectories – these are the locations where the characteristics cross and ∇u is undefined. Thus, a classical solution to eq. (4.2) usually does not exist. The theory of *viscosity solutions* introduced by Crandall and Lions [25] overcomes this difficulty by selecting the

unique weak solution coinciding with the value function of the original control problem. Restricting the process dynamics to Ω is similarly handled by switching to domain-constrained viscosity solutions [6, 154].

To simplify the exposition, we focus on *isotropic* problems, where the observability K and the speed of motion f depend only on $\mathbf{x}$. In this case, we choose $A = \{\boldsymbol{a} \in \mathbb{R}^d \mid |\boldsymbol{a}| = 1\}$ and interpret $\boldsymbol{a}$ as the direction of motion. Then $K(\mathbf{x}, \boldsymbol{a}) = K(\mathbf{x})$ and $\boldsymbol{f}(\mathbf{x}, \boldsymbol{a}) = f(\mathbf{x})\boldsymbol{a}$, with f encoding the speed of motion through the point $\mathbf{x}$. In this case, the optimal direction is known analytically: $\boldsymbol{a}^* = -\nabla u/|\nabla u|$ and eq. (4.2) reduces to an *Eikonal equation*

$$|\nabla u(\mathbf{x})| f(\mathbf{x}) = K(\mathbf{x}), \qquad \forall \mathbf{x} \in \Omega \setminus \{\mathbf{x}_T\}; \qquad\qquad u(\mathbf{x}_T) = 0. \tag{4.3}$$

The characteristics of these static PDEs are precisely the optimal trajectories, which define the direction of "information flow". This is quite useful once (4.3) is discretized on a grid (e.g., substituting upwind divided differences for partial derivatives, while taking $u = +\infty$ for all gridpoints outside of Ω to enforce the state constraints). The discretization yields a large coupled system of nonlinear equations. Knowing the characteristic direction for every gridpoint, one could, in principle, re-order the equations, effectively decoupling this system. But since the PDE is nonlinear, its characteristic directions are not known in advance. One path[1] to computational efficiency is to determine those characteristic directions simultaneously with solving the discretized system, in the spirit of Dijkstra's classical algorithm for finding shortest paths on graphs [35]. Two such non-iterative methods (Tsitsiklis' Algorithm [170] and Sethian's Fast Marching

[1] Fast Sweeping [186] is another popular approach for gaining efficiency in solving Eikonal equations. We refer readers to [16,17] for a review of many other "fast" techniques, including the hybrid marching/sweeping methods that aim to combine the best features of both approaches. Even though our own implementation is based on Fast Marching, any of these methods can be used to solve isotropic control problems arising in subsequent sections. Which one will be faster depends on the domain geometry and the particular pointwise observability functions.

Method [144]) are applicable to this special isotropic case. Once eq. (4.3) is solved, the optimal trajectory may be recovered by finding the path orthogonal to the level curves of $u(x)$. This can be achieved numerically by the steepest descent method on $u(x)$. An example of the solution of eq. (4.3) is shown in 4.1.

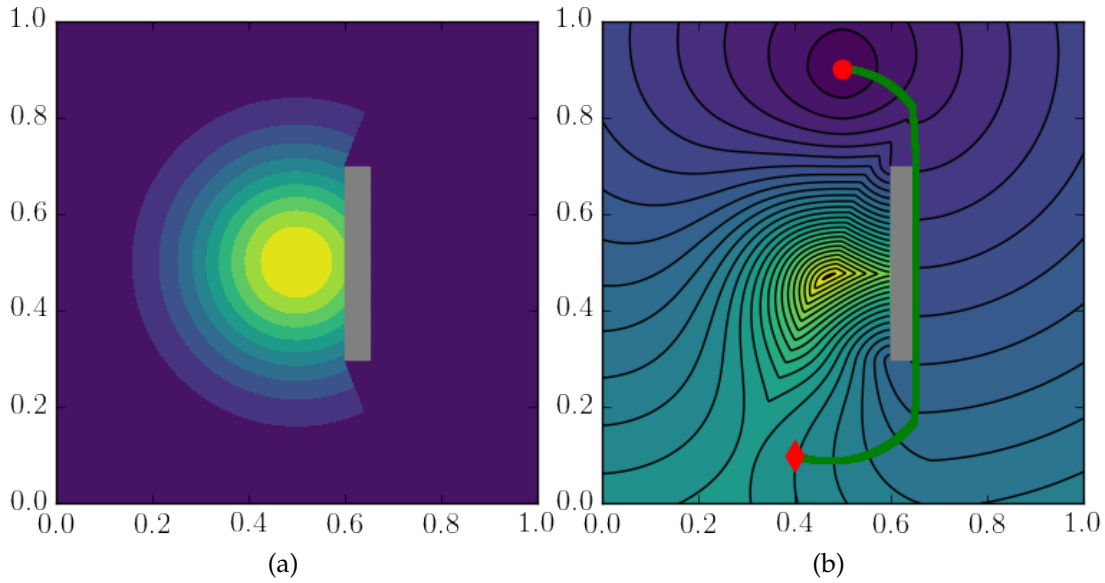


Figure 4.1: (a) The observability function $K(\mathbf{x})$ for an observer position $(0.5, 0.5)$. The gray rectangle is an obstacle, which obstructs the vision of the observer. The shadow zones created by the obstacle can be computed using the solution of the Eikonal equation (see section 4.5.1). (b) A contour plot of the solution of eq. (4.3) for $f(\mathbf{x}) = 1$ and the cost function in (a). The red diamond is the starting position, the red circle is the target position, and the green curve is the optimal trajectory, which is orthogonal to the level curves of $u(x)$ and follows a part of the obstacle boundary. See section 4.5 for additional information and parameters used.

4.3 Multiple observer locations and different notions of optimality

We now transition to the setting where the observer has a choice of multiple surveillance plans. Assuming that the observer remains stationary, this is equivalent to choosing its position from a fixed set of r locations $\mathcal{X} = \{\hat{\mathbf{x}}_1, \cdots, \hat{\mathbf{x}}_r\}$

known to the evader. Each location is associated with a pointwise observability function $K_i(\mathbf{x})$ for an evader moving through $\mathbf{x} \in \Omega$ and an observer stationed at $\hat{\mathbf{x}}_i$. (Typically, K_i is a decreasing function of $|\mathbf{x} - \hat{\mathbf{x}}_i|$ when $\mathbf{x}$ is visible from $\hat{\mathbf{x}}_i$ or a small constant $\sigma > 0$ if $\mathbf{x}$ is in a "shadow zone"; see section 4.5 for further details.) This results in r different definitions of the cumulative observability $\mathcal{J} = [\mathcal{J}_1, \ldots, \mathcal{J}_r]^T$ for a particular control. Ideally, a rational evader would prefer a path that minimizes its exposure to all possible observer locations $\hat{\mathbf{x}}_i$. Unfortunately, there usually does not exist a single control minimizing all $\mathcal{J}_i$'s simultaneously. This naturally leads us to a notion of Pareto optimal trajectories and the methods for computing them efficiently. We review two such methods[2] in section 4.3.1 and explain how they can be used for planning by an evader optimizing either the worst-case or average-case observability in section 4.3.2.

4.3.1 Multiobjective path planning

For a fixed starting position $\mathbf{x} \in \Omega$, a control $\boldsymbol{a}(\cdot)$ is *dominated* by a control $\hat{\boldsymbol{a}}(\cdot)$ if $\mathcal{J}_i(\mathbf{x}, \hat{\boldsymbol{a}}(\cdot)) \leq \mathcal{J}_i(\mathbf{x}, \boldsymbol{a}(\cdot))$ for all i and the inequality is strict for at least one of them. We call $\boldsymbol{a}(\cdot)$ *Pareto optimal* if it is not dominated by any other control. In other words, Pareto optimal controls are the ones that cannot be improved with respect to any one criterion without making them worse with respect to another. The vector of costs associated with each Pareto optimal control defines a point in $\mathbb{R}^r$ and the set of all such points is the Pareto Front (PF). In path planning applications, the PF is typically used to carefully evaluate all tradeoffs. (E.g., what is the smallest attainable $\mathcal{J}_1$ given the desired upper bounds on $\mathcal{J}_2, \ldots, \mathcal{J}_r$

[2] Here we describe these methods in terms of exposure to different observer's positions, but both of them were introduced for much more general multi-objective control problems. In many applications it is necessary to balance completely different criteria; e.g., time vs fuel vs money vs threat, etc. Other methods for approximating the full PF can be found in [61] and [33].

?)

Mitchell and Sastry developed a method for multiobjective path planning [113] based on the usual *scalarization* approach to multiobjective optimization [107]. Let $\Delta_r = \{\boldsymbol{\lambda} = (\lambda_1, \ldots \lambda_r) \mid \sum_{i=1}^r \lambda_i = 1, \text{ and all } \lambda_i \geq 0\}$. For each $\boldsymbol{\lambda} \in \Delta_r$ one can define a new pointwise observability function $K^{\boldsymbol{\lambda}} = \sum_{i=1}^r \lambda_i K_i$ and a new cumulative observability function $\mathcal{J}^{\boldsymbol{\lambda}} = \sum_i \mathcal{J}_i$. A weighted cost Eikonal equation

$$|\nabla u^{\boldsymbol{\lambda}}(\mathbf{x})| f(\mathbf{x}) = K^{\boldsymbol{\lambda}}(\mathbf{x}) \tag{4.4}$$

is then solved for a fixed $\boldsymbol{\lambda}$, providing a control function $\boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot)$ satisfying $\boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot) \in \arg\min_{\boldsymbol{a}(\cdot)\in\mathcal{A}} \mathcal{J}^{\boldsymbol{\lambda}}(\mathbf{x}_S, \boldsymbol{a}(\cdot))$. We call such a control function $\boldsymbol{\lambda}$-optimal. If $\lambda_i > 0$ for all i, the obtained $\boldsymbol{\lambda}$-optimal control is also guaranteed to be Pareto optimal; see fig. 4.2. However, if at least one $\lambda_i = 0$ and multiple $\boldsymbol{\lambda}$-optimal strategies exist for a specific $\boldsymbol{\lambda}$, then some of the $\boldsymbol{\lambda}$-optimal strategies may not be Pareto optimal. Such corner cases (illustrated in fig. 4.5) might require additional pruning; alternatively, one can rule out such non-Pareto trajectories by perturbing $\boldsymbol{\lambda}$ to ensure that all components are positive.

Additional linear PDEs can be solved simultaneously to compute the individual costs $(\mathcal{J}_1, \ldots \mathcal{J}_r)$ incurred along any $\boldsymbol{\lambda}$-optimal trajectory; e.g., when f and all K_i's are isotropic, the corresponding linear equations are

$$\nabla v_i^{\boldsymbol{\lambda}} \cdot \nabla u^{\boldsymbol{\lambda}} = K_i K^{\boldsymbol{\lambda}} / f^2, \tag{4.5}$$

where $v_i^{\boldsymbol{\lambda}}(\mathbf{x}) = \mathcal{J}_i\left(\mathbf{x}, \boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot)\right)$.

To approximate the PF, this procedure is repeated for a large number of $\boldsymbol{\lambda} \in \Delta_r$. Unfortunately, as shown in fig. 4.2, scalarization-based methods can only recover the convex portion of PF [28]. This is an important drawback since

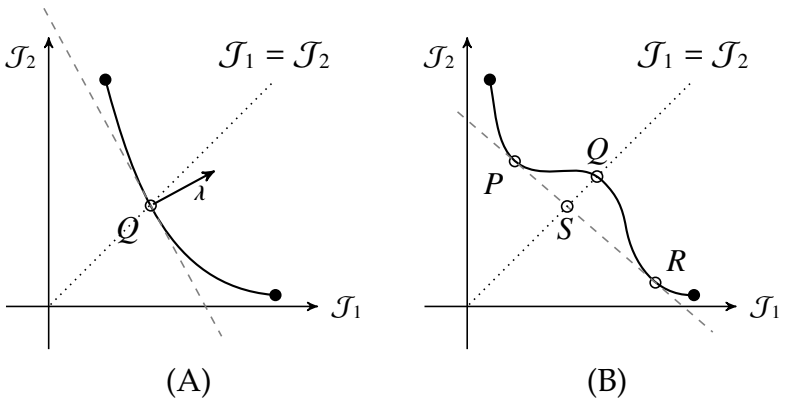


Figure 4.2: (A) Convex smooth Pareto Front with a point Q corresponding to the worst case optimal $\boldsymbol{\lambda} = (\lambda_1, \lambda_2) \in [0,1]^2$. The line perpendicular to $\boldsymbol{\lambda}$ is tangent to PF at Q. If any part of PF fell below it, the path corresponding to Q would not be $\boldsymbol{\lambda}$-optimal. The dotted line is the central ray (where $\mathcal{J}_1 = \mathcal{J}_2$)). (B) Non-convex smooth Pareto Front. Points P and R correspond to 2 different $\boldsymbol{\lambda}$-optimal paths. The portion of PF between P and R (including the worst-case optimal point Q) cannot be found by scalarization. Point S, found as a convex combination of P and R, is average-case optimal.

in many optimal control problems the non-convex parts of PF are very common and equally important. An alternative approach was developed in [89] to address this limitation and produce the entire PF for all $\mathbf{x} \in \Omega$ simultaneously. The method is applicable for any number of observer positions, but to simplify the notation we explain it here for $r = 2$ only. We expand the state space to $\Omega_e = \Omega \times [0, B]$ and define the new value function $w(\mathbf{x}, b) = \inf \mathcal{J}_1(\mathbf{x}, \boldsymbol{a}(\cdot))$, with the infimum taken over all controls satisfying $\mathcal{J}_2(\mathbf{x}, \boldsymbol{a}(\cdot)) \leq b$. Thus, b is naturally interpreted as the current "budget" for the secondary criterion. The value function is then recovered by solving an augmented PDE

$$\min_{\boldsymbol{a} \in A} \left\{ K_1(\mathbf{x}, \boldsymbol{a}) + \nabla_{\mathbf{x}} w \cdot \boldsymbol{f}(\mathbf{x}, \boldsymbol{a}) - K_2(\mathbf{x}, \boldsymbol{a}) \frac{\partial w}{\partial b} \right\} = 0. \tag{4.6}$$

The method in [89] uses a first-order accurate semi-Lagrangian discretization [45] to compute the discontinuous viscosity solution of (4.6) for a range of problems in multi-criterion path planning. The method was later generalized to treat constraints on reset-renewable resources [159]. The same approach was also

adapted to Probabilistic RoadMap graphs and field-tested on robotic platforms at the United Technologies Research Center [22].

Aside from approximating the entire PF, the key computational advantage is the *explicit causality*: since K_2 is positive, all characteristics are monotone in b and methods similar to the explicit "forward marching" in b-direction are applicable. (I.e., the system of discretized equations is trivially de-coupled.) Of course, the main drawback of the above idea is the higher dimensionality of Ω_e. For r observer locations, the scalarization approach [113] requires solving $(r+1)$ PDEs on $\Omega \subset \mathbb{R}^d$, but the parameter space Λ_r is $(r-1)$-dimensional. In contrast, with $w(\mathbf{x}, b)$ there are no parameters, but the computational domain is $(d+r-1)$-dimensional. Several techniques for restricting the computations to a relevant part of Ω_e were developed in [89], but the computational cost and memory requirements are still significantly higher than for any (single) HJB-solve in Ω.

4.3.2 Different notions of adversarial optimality

The Pareto Front allows us to answer one version of the surveillance-evasion problem: if the evader is completely risk-averse, he may choose the *worst-case optimal* strategy. That is, E will pick a control $\boldsymbol{a}_W(\cdot)$ that minimizes the observability from its "worst" observer position $\hat{\mathbf{x}}_i$:

$$\max_{\hat{\mathbf{x}}_i \in \mathcal{X}} \mathcal{J}_i(\boldsymbol{a}_W(\cdot)) \;\leq\; \max_{\hat{\mathbf{x}}_i \in \mathcal{X}} \mathcal{J}_i(\boldsymbol{a}(\cdot)), \qquad \forall \boldsymbol{a}(\cdot) \in \mathcal{A}.$$

This corresponds to the version of the problem where E is forced to "go first", with O selecting the maximizing $\hat{\mathbf{x}}_i \in \mathcal{X}$ in response. The following result shows that the intersection of Pareto Front with the "central ray" (i.e., the line where $\mathcal{J}_1 = \mathcal{J}_2 \cdots = \mathcal{J}_r$) yields the worst-case optimal strategy for E:

Theorem 4.3.1. *If $\boldsymbol{a}_{=}(\cdot)$ is a Pareto-optimal control satisfying $\mathcal{J}_i(\boldsymbol{a}_{=}(\cdot)) = \mathcal{J}_j(\boldsymbol{a}_{=}(\cdot))$ for all $i, j \in \{1, \cdots r\}$, then $\boldsymbol{a}_{=}(\cdot)$ is also worst-case optimal.*

Proof. Suppose there exists $\boldsymbol{a}'(\cdot)$ s.t.

$$\max_{\hat{\mathbf{x}}_i \in \mathcal{X}} \mathcal{J}_i(\boldsymbol{a}'(\cdot)) \; < \; \max_{\hat{\mathbf{x}}_i \in \mathcal{X}} \mathcal{J}_i(\boldsymbol{a}_{=}(\cdot))$$

then for all j we have:

$$\mathcal{J}_j(\boldsymbol{a}'(\cdot)) \leq \max_{\hat{\mathbf{x}}_i \in \mathcal{X}} \mathcal{J}_i(\boldsymbol{a}'(\cdot)) < \max_{\hat{\mathbf{x}}_i \in \mathcal{X}} \mathcal{J}_i(\boldsymbol{a}_{=}(\cdot)) = \mathcal{J}_j(\boldsymbol{a}_{=}(\cdot)),$$

which contradicts the Pareto-optimality of $\boldsymbol{a}_{=}(\cdot)$. □

The corresponding vector of costs $\boldsymbol{\mathcal{J}}(\boldsymbol{a}_{=}(\cdot))$ may lie on the convex portion of PF, as in Figures 4.2(A) and 4.3, in which case $\boldsymbol{a}_W = \boldsymbol{a}_{=}$ can be found by scalarization [113]. But if $\boldsymbol{\mathcal{J}}(\boldsymbol{a}_{=}(\cdot))$ lies on the non-convex portion of PF, as in Figures 4.2(B) and 4.4, the computational cost of finding the evader's worst-case optimal strategy grows exponentially with r as it involves solving a non-linear differential equation in $(r+d-1)$ dimensions [89]. As it will be shown in sections 4.4-4.6, the latter scenario is particularly common on domains with obstacles.

Luckily, another interpretation of evader's objectives proves much more computationally tractable. Even though $\boldsymbol{a}_{=}(\cdot)$ yields the lowest *worst-case* observability that E can achieve if he must choose a single control function deterministically, E might be able to attain an even lower expected (or *average-case*) observability if he switches to "mixed policies", choosing a probability distribution over a set of Pareto optimal controls. This is illustrated in fig. 4.2(B): by choosing probabilistically a path corresponding to the point P and another corresponding to point R, E obtains a new point S on the central ray, whose expected observability is lower than for the worst-case optimal Q regardless of O's

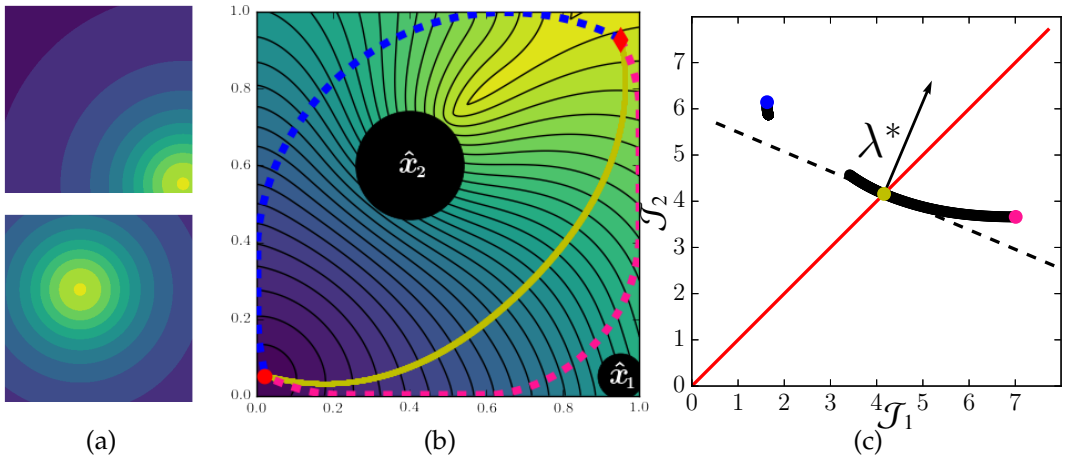


Figure 4.3: (a) Two observer positions and the corresponding observability maps K_i. (b) The λ^*-optimal path corresponding to $\lambda^* \approx (0.30, 0.70)$ is shown in yellow over the level sets of u^{λ^*}. The radii of black disks centered at $\hat{x}_i'$s are proportional to the corresponding components of λ^*. The two best response trajectories used when O chooses $\hat{\mathbf{x}}_1$ or $\hat{\mathbf{x}}_2$ are shown in blue and pink respectively. The trajectory in yellow is worst-case optimal for the evader as it is equally observable from both locations. (c) The convex part of Pareto Front (computed using the scalarization approach) intersects the "central ray" ($\mathcal{J}_1 = \mathcal{J}_2$, shown in red). The worst-case optimal vector λ^* is orthogonal to PF at the point of intersection (in yellow), whose coordinates correspond to the partial costs of the optimal path. The probability distribution λ^*, together with the yellow trajectory form a Nash equilibrium of the strategic game between the evader and the observer described in section 4.4. See section 4.5 for additional information and parameters used.

selected location. This, of course, assumes that O's location is selected without knowing in advance which of the two paths will be used by E. Indeed, for any single run from $\mathbf{x}_S$ to $\mathbf{x}_T$, the *worst-case* observability of this probabilistic policy is based on the worst cases for P and R, which (from the point of view of a completely risk-averse evader) would make the average-case optimal S inferior to the worst-case optimal Q. This scenario is fully realized in fig. 4.4, where $J_1(\boldsymbol{a}_=(\cdot)) = J_2(\boldsymbol{a}_=(\cdot)) \approx 4.94$, the expected observability corresponding to the optimal "probabilistic mix" of yellow and green trajectories is ≈ 4.83, but the worst case associated with this mixed policy is $J_1(\text{yellow}) \approx 6.03$.

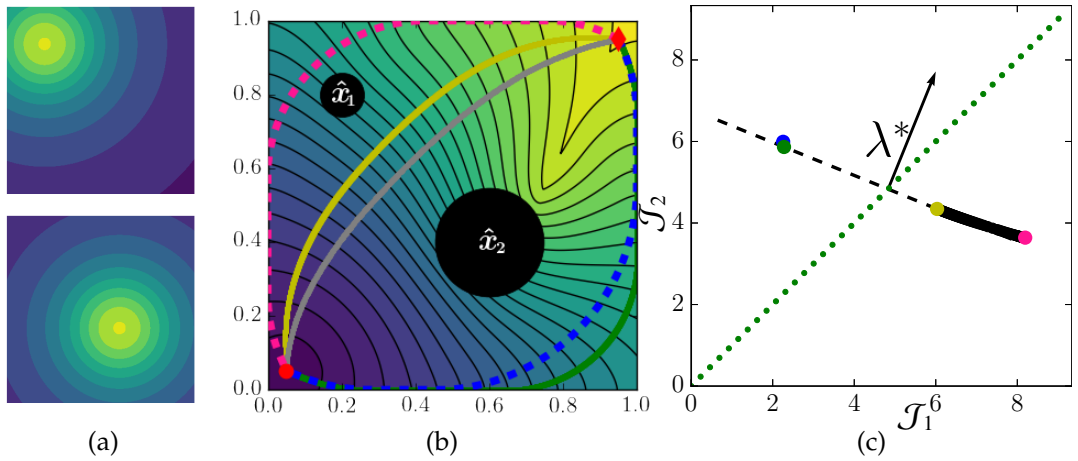


Figure 4.4: (a) Two observer positions and the corresponding observability maps K_i. (b) Two λ^*-optimal trajectories corresponding to $\lambda^* \approx (0.29, 0.71)$ are shown in yellow and green over the level sets of u^{λ^*}. The two best response trajectories used when O chooses $\hat{\mathbf{x}}_1$ or $\hat{\mathbf{x}}_2$ are shown in blue and pink respectively. The worst-case optimal trajectory is plotted in gray. (c) The convex part of the Pareto Front (in cyan) computed using the scalarization approach, and the whole Pareto Front (in black) computed using the method in [89]. The convex part of the Pareto Front does not intersect the central ray (shown in red). The worst-case optimal strategy (in gray) lies on the non-convex part of the Pareto Front and thus cannot be computed using scalarization. The Nash equilibrium pair of strategies consists of using positions $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ with probabilities λ^* for O and using the yellow and green trajectories (both of which lie on the convex part of the PF) with probability $\big[p(\text{yellow}), p(\text{green})\big] = [0.29, 0.71]$ for E (see section 4.4). The latter mixed strategy is average-case optimal for E. See section 4.5 for additional information and parameters used.

We note that O could also consider using a mixed strategy. In this case, K^λ can be interpreted as the expected pointwise observability when using the probability density $\lambda \in \Delta_r$ over the positions $\mathcal{X}$. Similarly $\mathcal{J}^\lambda(\boldsymbol{a}(\cdot))$ is the expected cumulative observability when using the control function $\boldsymbol{a}(\cdot)$. fig. 4.2 shows that when we are interested in the average-case optimal behavior for both O and E, we only need to consider a convex hull of PF (denoted *co*(PF)), and the scalarization is thus adequate. Note that in figs. 4.3, 4.4 and 4.6, the set *co*(PF) was approximated by imposing a fine grid on Δ_r and re-solving

eq. (4.4) for each sampled $\boldsymbol{\lambda}$. Since we only care about the intersection of co(PF) with the central ray, this procedure is wasteful – and prohibitively expensive for high r. In the next section, we consider the case where both E and O optimize the expected/average-case performance by reformulating this as a semi-infinite strategic zero-sum game. We show that such Surveillance-Evasion Games (SEGs) can be solved through scalarization combined with convex optimization, without approximating the (convex hull of the) entire Pareto Front.

Remark 1. *Up till now, our geometric interpretation in figs. 4.3, 4.4 and 4.6 assumed that either PF or at least the co(PF) must intersect the central ray. If this is not the case, O will avoid using some of his positions. E.g., fig. 4.5 shows the pink and yellow trajectories corresponding to* $\boldsymbol{a}_1(\cdot)$ *and* $\boldsymbol{a}_2(\cdot)$*, which are optimal with respect to the observer positions* $\hat{\mathbf{x}}_1$ *and* $\hat{\mathbf{x}}_2$*. Since* $\mathcal{J}_1(\boldsymbol{a}_2(\cdot)) \leq \mathcal{J}_2(\boldsymbol{a}_2(\cdot))$*, the E's worst-case for* $\boldsymbol{a}_2(\cdot)$ *is actually the observer location* $\hat{\mathbf{x}}_2$*. A generalization of this scenario for* $r > 2$ *is covered in theorem 4.4.2.*

4.4 Surveillance-Evasion Games (SEGs)

In this section, we reformulate the problem of evasive path planning under surveillance uncertainty as a strategic game. This can model either the actual adversarial interactions between two players or the risk-averse logic of the evader even if the surveillance patterns are not likely to change in response to that evader's strategy. (The latter case is typically interpreted as a "game against nature".)

We assume that the evader is attempting to minimize (while the observer is attempting to maximize) the total expected observability integrated over E's tra-

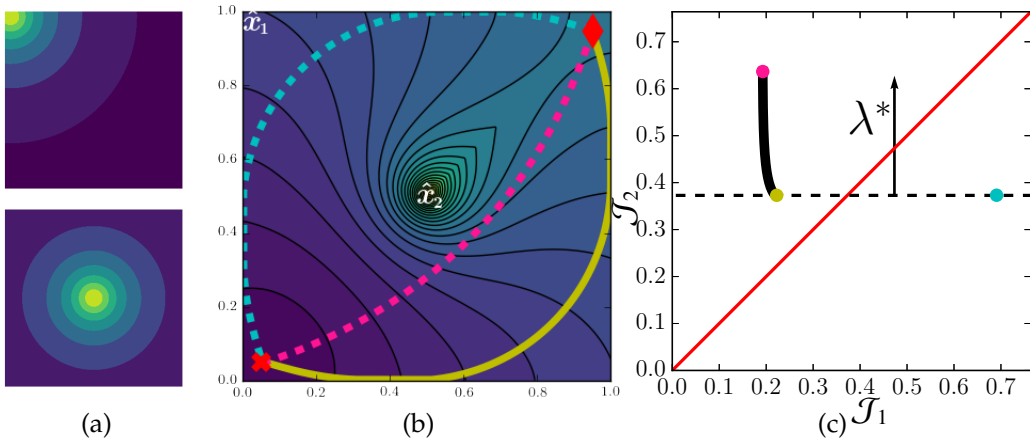


Figure 4.5: (a) Two observer positions and the corresponding observability maps K_i plotted in logarithmic scale. (b) The value function u^{λ^*} at $\lambda^* = (0, 1)$. The worst-case optimal strategy for O is the yellow trajectory, but both the yellow trajectory and the light blue trajectories are λ^*-optimal. The pink trajectory is the best response when the observer uses position $\hat{\mathbf{x}}_1$. (c) The Pareto Front does not intersect the central ray. The worst-case optimal trajectory is the one point on the Pareto Front that is closest to the central ray: the yellow point. The blue point is λ^*-optimal but it is not Pareto optimal as it is dominated by the yellow point. The Nash equilibrium strategy consists of the position $\hat{\mathbf{x}}_2$ for O, and the yellow trajectory for E (see section 4.4). See section 4.5 for additional information and parameters used.

jectories and dependent on O's positions. We further assume that O is aware of E's initial location $\mathbf{x}_S$ and its target location $\mathbf{x}_T$ but not of the trajectories chosen by E. Similarly, E is aware of the predefined locations of O, but not of the realized positions chosen by O. This game may be stated deterministically or stochastically. In the deterministic case, each player chooses a single pure strategy. That is, the observer chooses a single location $\hat{\mathbf{x}}_i \in \mathcal{X}$ and the evader chooses a single control function $\boldsymbol{a}(\cdot) \in \mathcal{A}$. In the probabilistic setting, each player chooses a mixed strategy, i.e., a probability distribution over the pure strategies. In other words, O chooses a probability distribution $\boldsymbol{\lambda} \in \Delta_r$ over positions and E chooses a probability distribution $\boldsymbol{\theta} \in \Delta_{\mathcal{A}}$ over control functions. The mixed strategy $\boldsymbol{\lambda}$ of the observer can be interpreted in several different ways:

1. O chooses a single position $\hat{\mathbf{x}}_i$ according to the probability distribution $\boldsymbol{\lambda}$ before E starts moving, and remains at that position until the end of the

round (that is, until E reaches the target).

2. O can randomly teleport between its positions at any time, and each λ_i reflects the proportion of time spent at the corresponding position $\hat{\mathbf{x}}_i$.
3. O has a budget of "observation resources", and $\boldsymbol{\lambda}$ reflects the fraction of these resources spent at each location. In this case, K_i reflects the pointwise observability corresponding to 100% of resources allocated to the position $\hat{\mathbf{x}}_i$.

Since we assume that neither player has access to the realization of the opponent's strategy in real time, these three interpretation are equivalent (and lead to the same optimal strategies) in our context. The payoff function of the game is the cumulative expected observability, and can be expressed as $P(\boldsymbol{\lambda}, \boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}}\left[\mathcal{J}^{\lambda}(\boldsymbol{a}(\cdot))\right]$ where $\mathbb{E}_{\boldsymbol{\theta}}[\cdot]$ denotes the expectation over the mixed strategy $\boldsymbol{\theta}$.

This SEG is a two-player *zero-sum game* [120], as each player's gains or losses are exactly balanced by the losses or gains of the opponent. Furthermore, it is *semi-infinite* as the set of pure strategies for O is a finite number r, whereas the set of pure strategies for E is uncountably infinite. A central notion of solution for strategic games is a Nash equilibrium [120], a pair of strategies for which neither player can improve his payoff by unilaterally changing his strategy. That is, a pair of strategies $(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*)$ is a Nash equilibrium if both of the following conditions hold:

$$\begin{aligned} P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*) &\leq P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}) \text{ for all } \boldsymbol{\theta} \in \Delta_{\mathcal{A}} , \\ P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*) &\geq P(\boldsymbol{\lambda}, \boldsymbol{\theta}^*) \text{ for all } \boldsymbol{\lambda} \in \Delta_r . \end{aligned} \tag{4.7}$$

A pure strategy Nash equilibrium does not always exist, therefore we focus on finding a mixed strategy Nash equilibrium. In our setting, the minimax

theorem for semi-infinite games [130] assures that a mixed strategy Nash equilibrium $(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*)$ exists, that all Nash equilibria have the same payoff, and that they are attained at the minimax (which is also equal to the maximin):

$$P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*) = \min_{\boldsymbol{\theta}\in\Delta_{\mathcal{A}}} \max_{\boldsymbol{\lambda}\in\Delta_r} \mathbb{E}_{\boldsymbol{\theta}}\left[\mathcal{J}^{\boldsymbol{\lambda}}(\boldsymbol{a}(\cdot))\right] = \max_{\boldsymbol{\lambda}\in\Delta_r} \min_{\boldsymbol{\theta}\in\Delta_{\mathcal{A}}} \mathbb{E}_{\boldsymbol{\theta}}\left[\mathcal{J}^{\boldsymbol{\lambda}}(\boldsymbol{a}(\cdot))\right] . \tag{4.8}$$

Although $\boldsymbol{\theta}$ is a probability distribution over the uncountable set $\Delta_{\mathcal{A}}$, there always exists an optimal mixed strategy $\boldsymbol{\theta}^*$ which is a mixture of at most r pure strategies, where r is the maximum number of positions for the observer [130]. In fact, it is easy to show that there will always exist a Nash equilibrium $(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*)$ with the number of pure strategies used in $\boldsymbol{\theta}^*$ not exceeding the number of non-zero components in $\boldsymbol{\lambda}^*$.

In the case of finite two-player zero-sum games, computing the Nash equilibrium is easily achieved by linear programming. For our SEGs, the challenge in computing a Nash equilibrium arises from enumerating the control functions $\boldsymbol{a}(\cdot) \in \mathcal{A}$ which are part of E's mixed strategy. Indeed, we do not possess a useful parametrization of the set of control functions $\mathcal{A}$, and our only computational kernel to generate a single $\boldsymbol{\lambda}$-optimal control function $\boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot)$ is to solve the weighted-cost Eikonal equation in eq. (4.4). For that reason, our solution strategy to compute the Nash Equilibrium involves two steps:

1. Find an approximate optimal strategy of the observer $\boldsymbol{\lambda}^*$ using convex optimization (see section 4.4.1).
2. Find an approximate optimal strategy of the evader $\boldsymbol{\theta}^*$ by generating near-optimal control functions (see section 4.4.2).

4.4.1 Optimal strategy of the Observer

In order to compute an optimal strategy $\boldsymbol{\lambda}^*$ of the observer, we consider the following problem:

$$\max_{\boldsymbol{\lambda}\in\Delta_r} \min_{\boldsymbol{a}(\cdot)\in\mathcal{A}} \mathcal{J}^{\boldsymbol{\lambda}}(\mathbf{x}_S, \boldsymbol{a}(\cdot)) = \max_{\boldsymbol{\lambda}\in\Delta_r} u^{\boldsymbol{\lambda}}(\mathbf{x}_S) \,. \tag{4.9}$$

For any fixed strategy $\boldsymbol{\lambda}$ for O, the inner minimization represents the optimal response of player E to that fixed strategy. Therefore, the maximin problem answers the question: what is the optimal strategy for O given that E gets to observe that strategy and respond? We call this problem the *E-response* problem. Note that although E could use a mixed strategy, there always exists a pure strategy which is optimal. That is:

$$\min_{\theta\in\Delta_{\mathcal{A}}} \mathbb{E}_\theta\left[\mathcal{J}^{\boldsymbol{\lambda}}(\boldsymbol{a}(\cdot))\right] = \min_{\boldsymbol{a}(\cdot)\in\mathcal{A}} \mathcal{J}^{\boldsymbol{\lambda}}(\boldsymbol{a}(\cdot)) \,. \tag{4.10}$$

This implies that any optimal $\boldsymbol{\lambda}$ for eq. (4.9) is also an optimal $\boldsymbol{\lambda}$ for eq. (4.8). Consequently, the optimal $\boldsymbol{\lambda}$ for eq. (4.9) is one half of a Nash equilibrium pair. However, the optimal pair $(\boldsymbol{\lambda}, \boldsymbol{a}(\cdot))$ of eq. (4.9) is not a Nash equilibrium, except in a specific situation described in the following theorem.

Theorem 4.4.1. *Suppose there exists $\boldsymbol{\lambda}_= \in \Delta_r$ with associated $\boldsymbol{\lambda}_=$-optimal control function $\boldsymbol{a}^{\boldsymbol{\lambda}_=}(\cdot)$ which satisfies $\mathcal{J}_i(\boldsymbol{a}^{\boldsymbol{\lambda}_=}(\cdot)) = \mathcal{J}_j(\boldsymbol{a}^{\boldsymbol{\lambda}_=}(\cdot))$ for all $i, j \in \{1, \ldots, r\}$, then $(\boldsymbol{\lambda}_=, \boldsymbol{a}^{\boldsymbol{\lambda}_=}(\cdot))$ is a Nash equilibrium.*

Proof. The fact that E cannot improve his payoff follows from the definition of $\boldsymbol{a}^{\boldsymbol{\lambda}_=} \in \arg\min_{\boldsymbol{a}(\cdot)} \mathcal{J}^{\boldsymbol{\lambda}_=}(\boldsymbol{a}(\cdot))$. O may not improve his payoff either as for all $\boldsymbol{\lambda}$,

$$\mathcal{J}^{\boldsymbol{\lambda}}(\boldsymbol{a}^{\boldsymbol{\lambda}_=}(\cdot)) = \sum \boldsymbol{\lambda}_i \mathcal{J}_i(\boldsymbol{a}^{\boldsymbol{\lambda}_=}(\cdot)) = \sum \boldsymbol{\lambda}_{=,i} \mathcal{J}_i(\boldsymbol{a}^{\boldsymbol{\lambda}_=}(\cdot)) = \mathcal{J}^{\boldsymbol{\lambda}_=}(\boldsymbol{a}^{\boldsymbol{\lambda}_=}(\cdot)) \,.$$

□

This situation corresponds to the case when the convex part of the Pareto Front intersects the central ray, such as in the example in fig. 4.3. Theorem 4.3.1 implies that in this case, the worst-case optimal strategy for E coincides with E's half of the Nash equilibrium. Note that in general such a $\boldsymbol{\lambda}_{=}$ does not have to exist; e.g., in fig. 4.4 and fig. 4.6 the convex part of the Pareto Front does not intersect the central ray. In such situations, the worst-case optimal strategy for E and the Nash Equilibrium are different. Moreover, the latter involves a mixed strategy for E covered in section 4.4.2.

We now direct our attention to solving the E-response problem numerically. Equation (4.9) may be stated as the following optimization problem:

$$\begin{aligned} &\max_{\boldsymbol{\lambda}} G(\boldsymbol{\lambda}) \\ &\text{s.t. } \lambda_i \geq 0, \quad \sum_{i=1}^{r} \lambda_i = 1 \ . \end{aligned} \tag{4.11}$$

The objective function $G(\boldsymbol{\lambda}) = \min_{\boldsymbol{a}(\cdot)\in\mathcal{A}} \sum_{i=1}^{r} \boldsymbol{\lambda}_i \mathcal{J}_i(\boldsymbol{a}(\cdot))$ is concave as it is the pointwise minimum of linear functions. Furthermore, the vector of individual cumulative costs $\boldsymbol{\mathcal{J}}(\boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot))$, where $\boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot) \in \arg\min_{\boldsymbol{a}(\cdot)\in\mathcal{A}} \mathcal{J}^{\boldsymbol{\lambda}}(\boldsymbol{a}(\cdot))$, is a supergradient of G (denoted as $\boldsymbol{\mathcal{J}}(\boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot)) \in \partial G(\boldsymbol{\lambda})$). A supergradient provides an ascent direction of a concave function, i.e., $\boldsymbol{w} \in \partial G(\boldsymbol{\lambda})$ if for all $\hat{\boldsymbol{\lambda}} \in \Delta_r$,

$$G(\hat{\boldsymbol{\lambda}}) - G(\boldsymbol{\lambda}) \leq \boldsymbol{w}^T(\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}) \ .$$

The fact that $\mathcal{J}(\boldsymbol{a}^{\lambda}(\cdot)) \in \partial G(\boldsymbol{\lambda})$ is seen from the following computation: for any $\hat{\boldsymbol{\lambda}}$,

$$\begin{aligned} G(\hat{\boldsymbol{\lambda}}) - G(\boldsymbol{\lambda}) &= \left(\min_{\boldsymbol{a} \in \mathcal{A}} \sum_{i=1}^{r} \hat{\lambda}_i \mathcal{J}_i(\boldsymbol{a}(\cdot)) \right) - \sum_{i=1}^{r} \lambda_i \mathcal{J}_i(\boldsymbol{a}^{\lambda}(\cdot)) \\ &\leq \sum_{i=1}^{r} \hat{\lambda}_i \mathcal{J}_i(\boldsymbol{a}^{\lambda}(\cdot)) - \sum_{i=1}^{r} \lambda_i \mathcal{J}_i(\boldsymbol{a}^{\lambda}(\cdot)) \\ &= \mathcal{J}(\boldsymbol{a}^{\lambda}(\cdot))^T (\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}) \, . \end{aligned}$$

Evaluating the vector $\mathcal{J}(\boldsymbol{a}^{\lambda}(\cdot))$ can be challenging computationally; we show how this can be done in section 4.5.2. Once this ascent direction is known, one still needs to ensure that $\boldsymbol{\lambda}$ remains a feasible probability distribution over $\mathcal{X}$, and we use the orthogonal projection operator $\Pi : \mathbb{R}^r \to \Delta_r$. The operator Π can be computed in $O(r \log r)$ operations [12,174] as summarized in algorithm 4. The resulting projected supergradient method [8, Chap. 8] is shown in algorithm 5. The iterates of algorithm 5 for the example from fig. 4.6 are illustrated in fig. 4.7.

Algorithm 4 Orthogonal projection onto the probability simplex

Input $\boldsymbol{\lambda} \in \mathbb{R}^r$
Sort $\boldsymbol{\lambda}$ into $\boldsymbol{u}$: $u_1 \geq u_2 \geq \cdots \geq u_r$
Find $\rho = \max\{1 \leq j \leq r : u_j + \frac{1}{j}\left(1 - \sum_{i=1}^{j} u_i\right) > 0\}$
$\tau \leftarrow \frac{1}{\rho}\left(1 - \sum_{i=1}^{\rho} u_i\right)$
return x s.t. $x_i = \max\{\lambda_i + \tau, 0\}$, $i = 1, \ldots r$.

Algorithm 5 Projected supergradient method for finding the maximum of G over the set Δ_r

Input Initial guess $\boldsymbol{\lambda}_0$, stepsizes α_k, number of iterations n
for $k = 0 : (n-1)$ **do**
    Compute $G(\boldsymbol{\lambda}_k) = u^{\boldsymbol{\lambda}_k}(\mathbf{x}_S)$ and find some $\boldsymbol{g} \in \partial G(\boldsymbol{\lambda}_k)$
    $\boldsymbol{\lambda}_{k+1} \leftarrow \Pi(\boldsymbol{\lambda}_k + \alpha_k \boldsymbol{g})$
end for
return $\arg\max_{\boldsymbol{\lambda} \in \{\boldsymbol{\lambda}_0, \ldots, \boldsymbol{\lambda}_n\}} G(\boldsymbol{\lambda})$

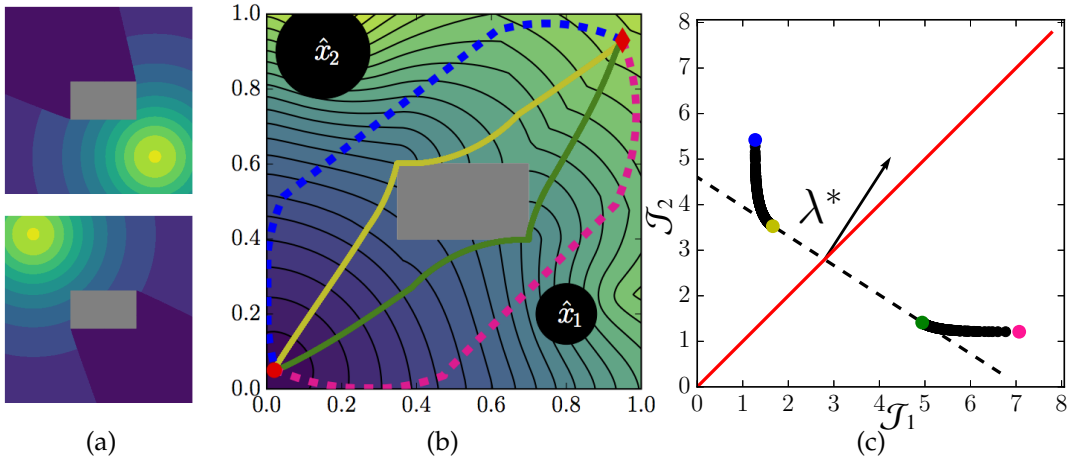


Figure 4.6: (a) Two observer positions and the corresponding observability maps K_i on a domain with a single obstacle (shown in gray). (b) Two λ^*-optimal trajectories corresponding to $\lambda^* \approx (0.39, 0.61)$ are shown in yellow and green over the level sets of u^{λ^*}. The two best response trajectories used when O chooses $\hat{\mathbf{x}}_1$ or $\hat{\mathbf{x}}_2$ are shown in blue and pink respectively. The trajectories in yellow and green are not worst-case optimal for the evader but are used in E's mixed Nash equilibrium strategy. (c) The convex part of the Pareto Front does not intersect the central ray (shown in red). This is the same situation already observed in fig. 4.4, but it is even more common on domains with obstacles. The Nash equilibrium pair of strategies consists of using positions $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ with probabilities λ^* for O, and using the yellow and green trajectories with probability $\big[p(\text{yellow}), p(\text{green})\big] = [0.65, 0.35]$ for E. See section 4.5 for additional information and parameters used.

4.4.2 Optimal strategy of the Evader

Computing the evader's half of the Nash equilibrium is more challenging due to the fact that the set of E's pure strategies, i.e., the set of control functions $\boldsymbol{a}(\cdot)$ leading from the source $\mathbf{x}_S$ to the target $\mathbf{x}_T$, is uncountably infinite. We propose a heuristic strategy to approximate $\boldsymbol{\theta}^*$ which relies on two properties of the Nash equilibrium in semi-infinite games:

1. There exists a Nash mixed strategy for E which uses only r pure strate-

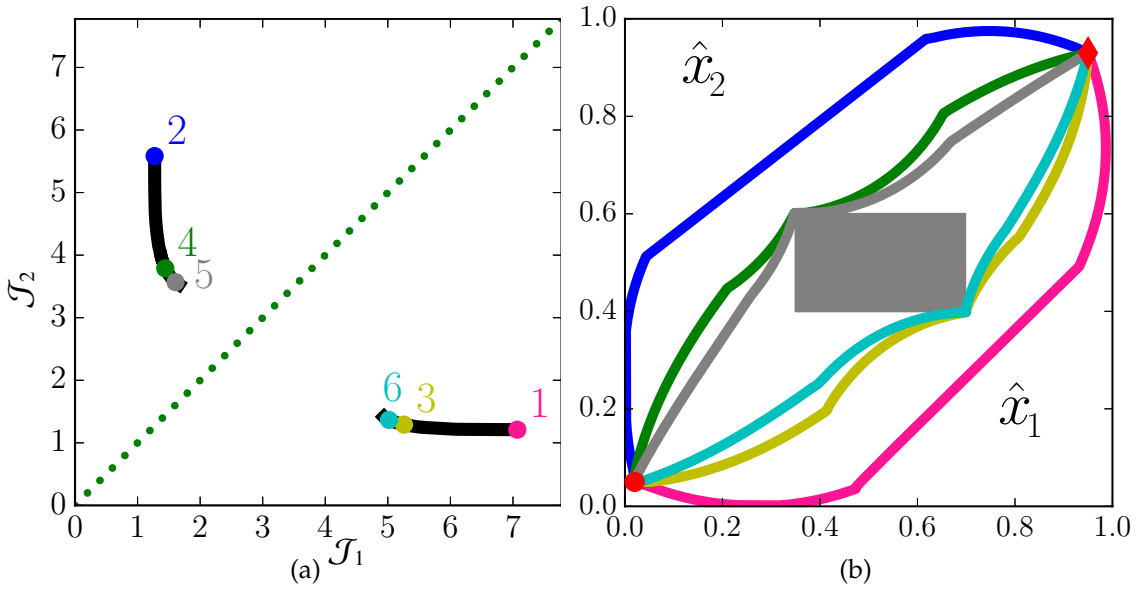


Figure 4.7: (a) Convex part of PF, and the individual costs of the first 6 iterates $\boldsymbol{\lambda}_k$ of algorithm 5 (with stepsizes $\alpha_k = 3/k$) for the problem shown in fig. 4.6. (b) The $\boldsymbol{\lambda}_k$-optimal trajectories of the first 6 iterates. We note that only a few iterates are needed to obtain trajectories which are close to the central ray. Thus, it does not require computing the whole PF which saves computational time.

gies[3] [130].

2. All pure strategies employed with positive probability in the Nash equilibrium have the same expected payoff, with the expectation taken over the other half of the Nash. In particular, all control functions used with positive probability in the Nash equilibrium are $\boldsymbol{\lambda}^*$-optimal.

The following characterization of the Nash equilibrium helps us generate a good candidate set of $\boldsymbol{\lambda}^*$-optimal trajectories.

Theorem 4.4.2. *Let* $(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*) \in \Delta_r \times \Delta_{\mathcal{A}}$ *and* $\mathcal{I} = \{i \mid \lambda_i^* \neq 0\}$. $(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*)$ *is a Nash equilibrium if and only if the following three conditions hold:*

1. $\boldsymbol{\lambda}^*$ *is a constrained maximizer of* $G(\boldsymbol{\lambda})$ *in eq.* (4.11),

[3]This result assumes that the set $S = \{(s_1, s_2, \ldots s_r) \mid s_i = P(\hat{\mathbf{x}}_i, \boldsymbol{a}(\cdot)); i = 1, 2, \ldots, r; \boldsymbol{a}(\cdot) \in \mathcal{A}\} \subset \mathbb{R}^r$ is bounded and $co(S)$ is closed. In our case, S is not bounded for the full set of control functions in $\mathcal{A}$ but becomes bounded if we restrict our attention to Pareto optimal control functions.

2. *if* $i \in \mathcal{I}$ *then* $\mathbb{E}_{\boldsymbol{\theta}^*}[\mathcal{J}_i(\boldsymbol{a}(\cdot))] = G(\boldsymbol{\lambda}^*)$, *and*

3. *if* $i \notin \mathcal{I}$, *then* $\mathbb{E}_{\boldsymbol{\theta}^*}[\mathcal{J}_i(\boldsymbol{a}(\cdot))] \leq G(\boldsymbol{\lambda}^*)$.

Proof. ($\Rightarrow$)

Suppose $(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*)$ is a Nash equilibrium. Item 1 follows from the minimax theorem for semi-infinite game and eq. (4.10). Assume item 2 does not hold, then there must exist $i, j \in \mathcal{I}$ s.t. $\mathbb{E}_{\boldsymbol{\theta}^*}[\mathcal{J}_i(\boldsymbol{a}(\cdot))] > \mathbb{E}_{\boldsymbol{\theta}^*}\left[\mathcal{J}_j(\boldsymbol{a}(\cdot))\right]$. Consider the strategy $\hat{\boldsymbol{\lambda}} \in \Delta_r$:

$$\hat{\lambda}_k = \begin{cases} \lambda_i^* + \lambda_j^* & \text{if } k = i \\ 0 & \text{if } k = j \\ \lambda_k^* & \text{otherwise} \end{cases}.$$

Then we have that:

$$P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*) = \sum_{i=1}^{r} \lambda_i^* \mathbb{E}_{\boldsymbol{\theta}^*}[\mathcal{J}_i(\boldsymbol{a}(\cdot))] < \sum_{i=1}^{r} \hat{\lambda}_i \mathbb{E}_{\boldsymbol{\theta}^*}[\mathcal{J}_i(\boldsymbol{a}(\cdot))] = P(\hat{\boldsymbol{\lambda}}, \boldsymbol{\theta}^*).$$

This contradicts that $(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*)$ is a Nash equilibrium, thus item 2 must hold. A similar argument can be used to demonstrate item 3: assume there exists $i \notin \mathcal{I}$ with $\mathbb{E}_{\boldsymbol{\theta}^*}[\mathcal{J}_i(\boldsymbol{a}(\cdot))] > G(\boldsymbol{\lambda}^*)$. Let $j \in \mathcal{I}$ and consider the strategy $\hat{\boldsymbol{\lambda}}$:

$$\hat{\lambda}_k = \begin{cases} \lambda_j^* & \text{if } k = i \\ 0 & \text{if } k = j \\ \lambda_k^* & \text{otherwise} \end{cases}$$

Once again, this implies that $P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*) < P(\hat{\boldsymbol{\lambda}}, \boldsymbol{\theta}^*)$ which contradicts that $(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*)$ is a Nash equilibrium.

($\Leftarrow$) Assume items 1 to 3 hold and suppose there exists $\boldsymbol{\theta}$ s.t. $P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}) < P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*)$, then there must exist $\boldsymbol{a}(\cdot)$, used with non-zero probability in $\boldsymbol{\theta}$ such

that:

$$\mathcal{J}^{\lambda^*}(\boldsymbol{a}(\cdot)) < P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*) = G(\boldsymbol{\lambda}^*) \, .$$

This contradicts the definition of $G(\boldsymbol{\lambda}^*) = \arg\min_{\boldsymbol{a}(\cdot)\in\mathcal{A}} \mathcal{J}^{\lambda^*}(\boldsymbol{a}(\cdot))$. Thus, for all $\boldsymbol{\theta} \in \Delta_{\mathcal{A}}$ we have that:

$$P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*) \leq P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}) \, . \tag{4.12}$$

From items 2 and 3 it follows that for all $\boldsymbol{\lambda} \in \Delta_r$:

$$P(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*) = \sum_{i=1}^{r} \lambda_i^* \mathbb{E}_{\boldsymbol{\theta}^*} \left[\mathcal{J}_i(\boldsymbol{a}(\cdot))\right] \geq \sum_{i=1}^{r} \lambda_i \mathbb{E}_{\boldsymbol{\theta}^*} \left[\mathcal{J}_i(\boldsymbol{a}(\cdot))\right] = P(\boldsymbol{\lambda}, \boldsymbol{\theta}^*) \, . \tag{4.13}$$

Equations (4.12) and (4.13) imply that $(\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*)$ is a Nash equilibrium. □

Any mix of $\boldsymbol{\lambda}^*$-optimal trajectories forms a $\boldsymbol{\lambda}^*$-optimal strategy for the evader. However, that mix is part of a Nash equilibrium only if the observer has no incentive to change his strategy in response. Theorem 4.4.2 says that this is the case when the $\boldsymbol{\theta}^*$ defining the mix of individual observability of $\boldsymbol{\lambda}^*$-optimal trajectories lies on the central ray of the Pareto Front for a reduced problem. I.e., the PF for the SEG where the observer has a potentially smaller number of positions (the ones which are used with positive probability in $\boldsymbol{\lambda}^*$). This PF is in an s dimensional criterion space, where $s = |\mathcal{I}| \leq r$. In fig. 4.3, the number of observer positions is $r = 2$, and the dimension of the "reduced" problem is also $s = 2$ since both positions are used with positive probability. In this example, a single $\boldsymbol{\lambda}^*$-optimal trajectory exists and corresponds to the intersection of the central ray and the convex part of the PF. In the examples from fig. 4.4 and fig. 4.6, we still have $r = 2$ and $s = 2$, however there are two $\boldsymbol{\lambda}^*$-optimal trajectories. The Nash mixed strategy for E is thus obtained by finding a probability distribution $(\omega_1, \omega_2) \in \Delta_2$ over these two trajectories $(\boldsymbol{a}_1(\cdot), \boldsymbol{a}_2(\cdot))$ such that the linear combination of their individual costs lies on the central ray, i.e., such that $\omega_1 \mathcal{J}_1(\boldsymbol{a}_1(\cdot)) + \omega_2 \mathcal{J}_1(\boldsymbol{a}_2(\cdot)) = \omega_1 \mathcal{J}_2(\boldsymbol{a}_1(\cdot)) + \omega_2 \mathcal{J}_2(\boldsymbol{a}_2(\cdot))$. In the example from fig. 4.5,

$r = 2$ and $s = 1$. The PF of the reduced problem is a single point, and thus trivially lies on the "central ray", yielding a pure Nash equilibrium strategy for E. In section 4.5, we show additional examples with $r = 3$, $s = 3$, and $r = 6$, $s = 4$. Computationally, Theorem 4.4.2 means that if we are able to find a set of g $\boldsymbol{\lambda}^*$-optimal control functions $\mathcal{A}(\boldsymbol{\lambda}^*) = \{\boldsymbol{a}_j(\cdot)\}_{j=1}^{j=g}$, such that items 2 and 3 hold for some probability distribution $\boldsymbol{\omega} \in \Delta_g$, then $\boldsymbol{\lambda}^*$ is O's optimal response to $\boldsymbol{\omega}$ and we have found a Nash equilibrium pair. Note that the minimum number of trajectories g needed to form a Nash equilibrium is bounded above by s.

One remaining task is finding such a set $\mathcal{A}(\boldsymbol{\lambda}^*)$. Multiple $\boldsymbol{\lambda}^*$-optimal controls only exist if $\mathbf{x}_S$ lies on a shockline of $u^{\boldsymbol{\lambda}^*}$, where the gradient is undefined (e.g., the $\lim_{\mathbf{x}_i \to \mathbf{x}_S} \nabla u(\mathbf{x}_i)$ can be different depending on the sequence $\{x_i\}_i$). Numerically, our approximation of $u^{\boldsymbol{\lambda}^*}$ will yield a single upwind approximation of $\nabla u^{\boldsymbol{\lambda}^*}$, yielding a single $\boldsymbol{\lambda}^*$-optimal trajectory. As we show in fig. 4.8, multiple optimal trajectories might lie in the same upwind quadrant and any numerical implementation of gradient descent will find only one of them. (In theory, one can approximate the other by perturbing $\mathbf{x}_S$, but the direction of perturbation is unobvious, particularly when $\mathbf{x}_S$ lies on an intersection of multiple shocklines, which is surprisingly common in this application as we show in further sections.)

This challenge is even more pronounced because 5 yields *an approximate* value of $\boldsymbol{\lambda}^*$, since $\mathbf{x}_S$ will now be only *near* a shockline for some perturbed $\boldsymbol{\lambda}_\delta^* = \boldsymbol{\lambda}^* + \delta\boldsymbol{\lambda}$. The resulting single $\boldsymbol{\lambda}_\delta^*$-optimal control will be a reasonable approximate solution for the max-min problem, but can be arbitrarily far from the solution to a min-max problem (where O has a chance to switch to another strategy).

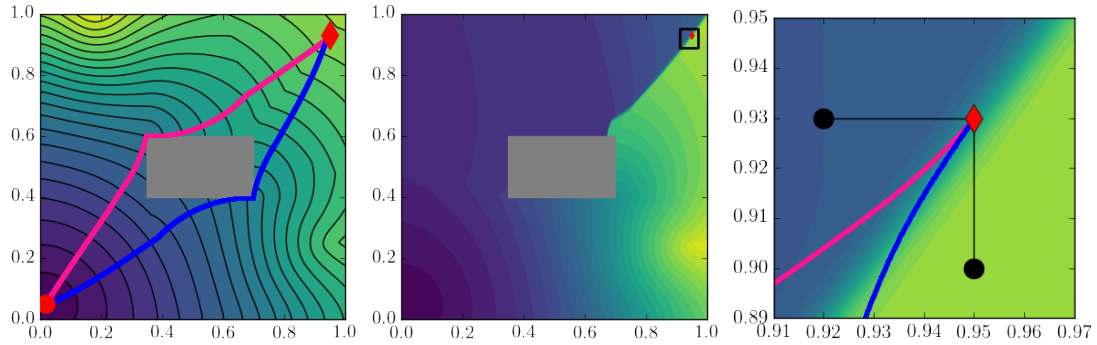


Figure 4.8: (a) Two λ^*-optimal trajectories in pink and blue plotted over the level sets of $u^{\lambda^*}(x)$. The source location $\mathbf{x}_S$ is on a shockline of $u^{\lambda^*}(x)$, the two trajectories have the same expected cumulative observability, but different individual cumulative observability. (b) The individual cost function $v_1^{\lambda^*}(x)$ is discontinuous at the source $\mathbf{x}_S$. The black square is the region displayed on (c). (c) The individual cost function $v_1^{\lambda^*}(x)$ zoomed in around the source and a depiction of the upwind stencil. The stencil (displayed larger for the sake of visualization) contains a point on either side of the line of discontinuity of $v_1^{\lambda^*}(x)$.

In view of these challenges, we opt for a different approach, where an approximation to $\mathcal{A}(\lambda^*)$ is computed iteratively, by adaptively growing a collection of λ^*_δ-optimal controls corresponding to different $\delta\lambda$'s. In some degenerate cases, generating even the first $\boldsymbol{a}_1(\cdot) \in \mathcal{A}(\lambda^*)$ may not be trivial since some λ^*-optimal control computed by solving the Eikonal will not be necessarily Pareto-optimal. E.g., in fig. 4.5 two control functions are λ^*-optimal, but only one of them is used in the Nash strategy of E as the blue trajectory violates item 3. However, both trajectories are indistinguishable from the point of view of the Eikonal solver since the position $\hat{\mathbf{x}}_1$ has zero weight in the weighted observability function K^{λ^*}. To address this issue whenever $s < r$, we set the weight of the pointwise observability of each unused position $i \notin \mathcal{I}$ to some small value ϵ (our implementation uses $\epsilon = 10^{-6}$). This is equivalent to seeking the solution of the weighted cost Eikonal equation for some perturbed $\lambda^*_\delta = (1-\epsilon)\lambda^* + \frac{\epsilon}{r-s} I_{\mathcal{I}^c}$, where $I_{\mathcal{I}^c}$ is the characteristic function of the complement of $\mathcal{I}$. We now turn our attention to finding further perturbations needed to generate λ^*_δ-optimal trajectories in order to

make item 2 approximately hold. Our goal is to have

$$\sum_{j=1}^{g} \omega_j \mathcal{J}_i(\boldsymbol{a}_j(\cdot)) = G(\boldsymbol{\lambda}^*) \tag{4.14}$$

approximately hold for all $i \in \mathcal{I} = \{i \mid \lambda_i^* > 0\}$. Unless this is already true with $g = 1$ (based on the previously found $\boldsymbol{a}_1(\cdot)$), we will need to find more $\boldsymbol{\lambda}_\delta^*$-optimal controls. Without loss of generality assume that $\mathcal{I} = \{1, \ldots, s\}$, and suppose we have already generated a set of k $\boldsymbol{\lambda}_\delta^*$-optimal trajectories $\mathcal{A}^k = \{\boldsymbol{a}_1(\cdot), \boldsymbol{a}_2(\cdot), \ldots, \boldsymbol{a}_k(\cdot)\}$, for some $k < g$. In order for eq. (4.14) to approximately hold, we will be increasing k until the norm of residual

$$\boldsymbol{R}(\boldsymbol{\omega}) = \begin{bmatrix} G(\boldsymbol{\lambda}^*) \\ G(\boldsymbol{\lambda}^*) \\ \vdots \\ G(\boldsymbol{\lambda}^*) \end{bmatrix} - \begin{bmatrix} \mathcal{J}_1(\boldsymbol{a}_1(\cdot)) & \mathcal{J}_1(\boldsymbol{a}_2(\cdot)) & \ldots & \mathcal{J}_1(\boldsymbol{a}_k(\cdot)) \\ \mathcal{J}_2(\boldsymbol{a}_1(\cdot)) & \mathcal{J}_2(\boldsymbol{a}_2(\cdot)) & \ldots & \mathcal{J}_2(\boldsymbol{a}_k(\cdot)) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{J}_s(\boldsymbol{a}_1(\cdot)) & \mathcal{J}_s(\boldsymbol{a}_2(\cdot)) & \ldots & \mathcal{J}_s(\boldsymbol{a}_k(\cdot)) \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_k \end{bmatrix} \tag{4.15}$$

falls under a threshold $\text{tol}_{\boldsymbol{R}}$. Assuming the set of trajectories $\mathcal{A}^k$ has already been computed, the probability distribution $\boldsymbol{\omega}^k \in \Delta_k$ minimizing the norm of this residual $\|\boldsymbol{R}(\boldsymbol{\omega}^k)\|_2$ can be found by quadratic programming. The residual vector $\boldsymbol{R}(\boldsymbol{\omega}^k)$ provides information about which control functions are missing. For example, consider the case where we observe that a single entry of $\boldsymbol{R}(\boldsymbol{\omega}^k)$ is large and positive, i.e., that for some $i \in \mathcal{I}$:

$$\sum_{j=1}^{k} \omega_j^k \mathcal{J}_i(\boldsymbol{a}_j(\cdot)) << G(\boldsymbol{\lambda}^*) \, .$$

The characterization in theorem 4.4.2 implies that $\mathcal{A}(\boldsymbol{\lambda}^*)$ should include at least one trajectory much more observable from position $\hat{\mathbf{x}}_i$. A $\boldsymbol{\lambda}_\delta^*$-optimal trajectory with that property can be found by perturbing $\boldsymbol{\lambda}$ to slightly decrease the role of $\hat{\boldsymbol{x}}_i$ in O's chosen strategy. This is equivalent to re-solving the Eikonal with $K^{\boldsymbol{\lambda}_\delta^*}$ corresponding to $\boldsymbol{\lambda}_\delta^* = \Pi_{\mathcal{I}}(\boldsymbol{\lambda}^* - \delta \boldsymbol{e}_i)$ where $\delta << 1$ is chosen adaptively (see algorithm 7), $\boldsymbol{e}_i$ is the i-th standard basis vector, and $\Pi_{\mathcal{I}}$ is the orthogonal projection

onto the simplex defined only with elements of $\mathcal{I}$. Once a new $\boldsymbol{\lambda}^*_\delta$-optimal control function has been found, we may solve the quadratic program in eq. (4.15) again with an additional column, and repeat the process until the norm of the residual is sufficiently small. More generally, a large $\|\boldsymbol{R}(\omega)\|$ implies that some control functions in $\mathcal{A}(\boldsymbol{\lambda}^*)$ (or some mix of control functions) not in the current set $\mathcal{A}^k$ has a high observability with respect to the positive entries of $\boldsymbol{R}(\omega)$ while having a low observability with respect to the negative entries of $\boldsymbol{R}(\omega)$. Thus, we set the perturbation direction to $-\boldsymbol{R}(\omega)$ instead of $-\boldsymbol{e}_i$. Throughout this perturbation step, the entries of $\boldsymbol{\lambda}^*$ associated with the complement $\mathcal{I}$ are held fixed. Our full method for computing an approximate Nash equilibrium is summarized in algorithm 6.

This method also has a geometric interpretation in terms of the Pareto Front. Whenever $\boldsymbol{\theta}^*$ is not a pure strategy, a hyperplane normal to $\boldsymbol{\lambda}^*$ supports PF at multiple points (corresponding to all controls in $\mathcal{A}(\boldsymbol{\lambda}^*)$). However, any generic perturbation of $\boldsymbol{\lambda}^*$ would result in a hyperplane supporting PF near only one of these points, and the approximation to $\boldsymbol{\lambda}^*$ found by algorithm 5, will correspond to a single optimal trajectory. For example, if we start with $\boldsymbol{a}_1(\cdot)$ corresponding to the yellow point in fig. 4.6c (and associated yellow trajectory in fig. 4.6b), then a small tilt (decreasing the role of position $\hat{\mathbf{x}}_1$ in O's plan) will yield a hyperplane supporting PF near the green point, allowing us to approximate the green trajectory in fig. 4.6b by solving the weighted cost Eikonal equation with observability function $K^{\boldsymbol{\lambda}^*_\delta}$.

Algorithm 6 Computing an approximate Nash equilibrium of the SEG.

Find $\boldsymbol{\lambda}^*$ using algorithm 5
$\mathcal{I} \leftarrow \{i \mid \lambda_i^* > \text{tol}_\lambda\}$
$\boldsymbol{\lambda}_\delta^* \leftarrow (1-\epsilon)\boldsymbol{\lambda}^* + \frac{\epsilon}{r-s} I_{\mathcal{I}^c}$
Find $\boldsymbol{\lambda}_\delta^*$-optimal control $\boldsymbol{a}_1(\cdot)$ and compute $\mathcal{J}_i(\boldsymbol{a}_1(\cdot))$ for all $i \in \mathcal{I}$
$k \leftarrow 1$, $\mathcal{A}^1 \leftarrow \{\boldsymbol{a}_1(\cdot)\}$, $\boldsymbol{\omega}^1 \leftarrow 1$
while $\|\boldsymbol{R}(\boldsymbol{\omega}^k)\| > \text{tol}_{\boldsymbol{R}}$ **do**
  $\boldsymbol{\lambda}_\delta^* \leftarrow (1-\epsilon)\Pi_{\mathcal{I}}\left(\boldsymbol{\lambda}^* - \delta \boldsymbol{R}(\boldsymbol{\omega}^k)\right) + \frac{\epsilon}{r-s} I_{\mathcal{I}^c}$
  Find $\boldsymbol{\lambda}_\delta^*$-optimal control $\boldsymbol{a}_{k+1}(\cdot)$ and compute $\mathcal{J}_i(\boldsymbol{a}_{k+1}(\cdot))$ for all $i \in \mathcal{I}$
  $\mathcal{A}^{k+1} \leftarrow \mathcal{A}^k \cup \{\boldsymbol{a}_k(\cdot)\}$
  $\boldsymbol{\omega}^{k+1} \leftarrow \arg\min_{\boldsymbol{\omega}^{k+1} \in \Delta_{k+1}} \|\boldsymbol{R}(\boldsymbol{\omega}^{k+1})\|_2$
  $k \leftarrow k+1$
end while
return $\boldsymbol{\lambda}^*$, $\mathcal{A}^k$, $\boldsymbol{\omega}^k$

4.5 Numerical matters

In this section, we detail the implementation of our algorithm and present additional numerical results. All algorithms were implemented in C++ and compiled with icpc version 16.0 on a MacBook Pro (16 GB RAM and an Intel Core i7 processor with four 2.5 GHz cores). The code is available online at `https://github.com/eikonal-equation/Stationary_SEG`. Our implementation relies on data structures and methods from Boost, Eigen and QuadProg++ libraries.

4.5.1 Functions, parameters, methods

All of our examples are posed on the domain $\Omega = [0,1]^2$ with the possible exclusion of obstacles. All figures are based on computations on a uniform cartesian grid of size $n \times n = 501 \times 501$ (with the grid spacing $h = 1/500$). To simplify the discussion, we always use a constant speed function $f(x) = 1$ though any

inhomogeneous speed can be similarly handled by solving the Eikonal equation eq. (4.3).

The pointwise observability functions are defined as

$$K_i(\mathbf{x}) = \begin{cases} \sigma, & \text{if } \mathbf{x} \text{ is in a shadow zone of } \hat{\mathbf{x}}_i; \\ \hat{K}(|\mathbf{x} - \hat{\mathbf{x}}_i|) + \sigma, & \text{otherwise.} \end{cases}$$

We set $\sigma = 0.1$ and $\hat{K}(r) = (\rho r^2 + 0.1)^{-1}$ with $\rho = 1$ in all examples except in fig. 4.5 (where we set $\rho = 30$ simply to improve the visualization). The visibility of each gridpoint with respect to each observer position is precomputed and stored, but the K_i values are computed on the fly as needed.

The shadow zones for each observer are precomputed as follows. For each observer location $\hat{\mathbf{x}}_i$, two distance functions are computed: $D_0^i(\mathbf{x})$ and $D^i(\mathbf{x})$. The first is the distance between $\hat{\mathbf{x}}_i$ and $\mathbf{x}$ when the obstacles are absent, while the second is that distance when obstacles are present. These distance functions can be computed by imposing the boundary conditions $D_0^i(\hat{\mathbf{x}}_i) = D^i(\hat{\mathbf{x}}_i) = 0$ and then solving two Eikonal equations [145]:

$$\left|\nabla D_0^i(\mathbf{x})\right| = 1, \qquad \left|\nabla D^i(\mathbf{x})\right| = \mathrm{Obs}(\mathbf{x}), \tag{4.16}$$

with $\mathrm{Obs}(\mathbf{x})$ set to ∞ inside the obstacles and 1 otherwise. The shadow zone of $\hat{\mathbf{x}}_i$ is characterized by $D^i > D_0^i$. But due to numerical errors in their approximation, we use a threshold value $\tau = 10^{-3}h$ (where h is the grid spacing) and specify that $\mathbf{x}$ is in this shadow zone whenever $D^i(\mathbf{x}) > D_0^i(\mathbf{x}) + \tau$.

The perturbation stepsize δ in algorithm 6 is chosen adaptively using algorithm 7. The goal of the adaptive strategy is to find the smallest perturbation δ necessary to obtain an additional $\boldsymbol{\lambda}_\delta^*$-optimal control function $\boldsymbol{a}_{k+1}(\cdot)$.

Algorithm 7 Adaptive strategy for choosing δ to generate $\boldsymbol{a}_{k+1}(\cdot)$

$\delta \leftarrow \delta_0$
$\boldsymbol{\lambda}^*_\delta \leftarrow (1-\epsilon)\Pi_{\mathcal{I}}\left(\boldsymbol{\lambda}^* - \delta \boldsymbol{R}(\boldsymbol{\omega}^k)\right) + \frac{\epsilon}{r-s} I_{\mathcal{I}^c}$
Compute a $\boldsymbol{\lambda}^*_\delta$-optimal control function $\hat{\boldsymbol{a}}(\cdot)$
while $\|\boldsymbol{\mathcal{J}}(\hat{\boldsymbol{a}}(\cdot)) - \boldsymbol{\mathcal{J}}(\boldsymbol{a}_j(\cdot))\|_2 < \text{tol}_\delta$ for any $j \in \{1, \dots k\}$ **do**
    $\delta \leftarrow 2\delta$
    $\boldsymbol{\lambda}^*_\delta \leftarrow (1-\epsilon)\Pi_{\mathcal{I}}\left(\boldsymbol{\lambda}^* - \delta \boldsymbol{R}(\boldsymbol{\omega}^k)\right) + \frac{\epsilon}{r-s} I_{\mathcal{I}^c}$
    Compute $\boldsymbol{\lambda}^*_\delta$-optimal control function $\hat{\boldsymbol{a}}(\cdot)$
end while
$\boldsymbol{a}_{k+1}(\cdot) \leftarrow \hat{\boldsymbol{a}}(\cdot)$

The initialization used in our implementation is $\delta_0 = 10^{-4}$, and the tolerance is set to $\text{tol}_\delta = 10^{-2}\|\boldsymbol{\mathcal{J}}(\hat{\boldsymbol{a}}(\cdot))\|_2$. The stepsize rule used in the supergradient iteration in algorithm 5 is $\alpha_k = 1/(k\|\boldsymbol{\mathcal{J}}(\boldsymbol{a}^{\boldsymbol{\lambda}_0}(\cdot))\|)$, the initial guess $\boldsymbol{\lambda}_0$ is a uniform distribution on $\mathcal{A}$ and the tolerance criteria on the residual and the near 0 entries used in algorithm 6 are $\text{tol}_{\boldsymbol{R}} = 10^{-3}G(\boldsymbol{\lambda}^*)$ and $\text{tol}_\lambda = 5 \cdot 10^{-3}$ respectively. The quadratic programming problem in eq. (4.15) is solved using the library QuadProg++.

4.5.2 Computation of individual costs

Running algorithm 5 requires computing the vector of individual observability $\boldsymbol{\mathcal{J}}(\mathbf{x}_S, \boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot))$. This problem is exactly the one solved by the scalarization approach described in section 4.3.1. Therefore, it can in principle be done by solving the Eikonal equation in eq. (4.3) with cost function $K^{\boldsymbol{\lambda}}$ and associated linear equations in eq. (4.5); i.e.: $G(\boldsymbol{\lambda}) = u^{\boldsymbol{\lambda}}(\mathbf{x}_S)$ and $\mathcal{J}_i(\mathbf{x}_S, \boldsymbol{a}^{\boldsymbol{\lambda}}(\cdot)) = v_i^{\boldsymbol{\lambda}}(\mathbf{x}_S)$. However, this technique has a severe drawback for this particular application: at the optimal $\boldsymbol{\lambda}^*$, $v_i^{\boldsymbol{\lambda}^*}$ is often discontinuous at $\mathbf{x}_S$. E.g., in fig. 4.8b, the upwind stencil containing the two $\boldsymbol{\lambda}^*$-optimal trajectories contains a point on either side of the

discontinuity line of $v_1^{\lambda^*}$ (which is the shockline of u^{λ^*}). As a result, the value of $v_1^{\lambda^*}(\mathbf{x}_S)$ is updated by interpolating the discontinuous function $v_1^{\lambda^*}$ across the line of discontinuity.

This effect happens when multiple trajectories are $\boldsymbol{\lambda}^*$-optimal. Each of these trajectories has the same expected cumulative observability $\mathcal{J}^{\lambda^*} = \sum_i \lambda_i^* \mathcal{J}_i$, but different individual observability $\mathcal{J}_i$. This issue leads to a large numerical error when using $v_i^{\lambda}(\mathbf{x}_S)$ to estimate the supergradient in algorithm 5, causing poor convergence of the method. Instead, we use the following process to compute the individual costs: first we solve the weighted cost Eikonal equation eq. (4.4) to obtain u^{λ} for a fixed $\boldsymbol{\lambda}$, then we trace the path $\mathbf{y}(t)$ using a gradient descent method on the value function u^{λ} and numerically estimate the integrals:

$$\mathcal{J}_i(\mathbf{x}_S, \boldsymbol{a}^{\lambda}(\cdot)) = \int_0^{T_{\boldsymbol{a}^{\lambda}}} K_i(\mathbf{y}(t), \boldsymbol{a}^{\lambda}(t))\, dt, \qquad i = 1, \ldots, r.$$

4.5.3 Additional experiments and error metrics

We present two additional examples that include a higher number of observer plans. In fig. 4.9, we show an example where the mixed strategy Nash equilibrium consists of a distribution over three strategies for both the evader and the observer. Figure 4.9 shows the value function u^{λ^*} at the optimal $\boldsymbol{\lambda}^*$. We observe that three shocklines of the value function u^{λ^*} meet at the source location $\mathbf{x}_S$, which implies that four trajectories are optimal starting from this location. However, the minimax theorem for infinite games assures that only 3 pure strategies are necessary to form a Nash equilibrium. Using algorithm 6, we find an approximate Nash equilibrium which uses a mix of such three trajectories.

In Figure 4.10, we show a maze-like example where the observer may choose

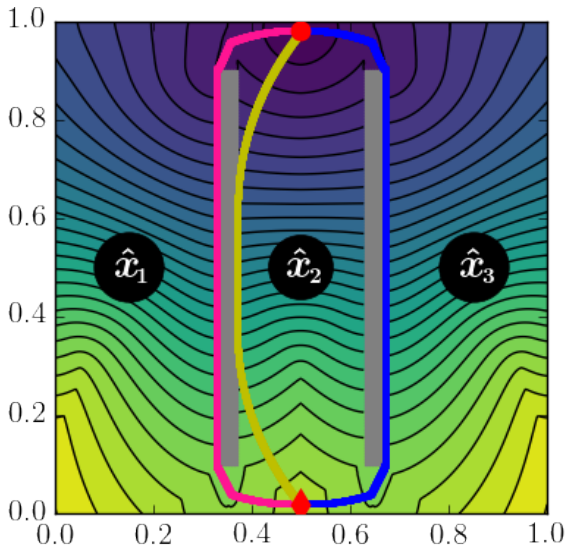


Figure 4.9: Computed Nash equilibrium for a situation where a mix of three pure strategies are necessary for each player. The value function u^{λ^*} with three near-λ^*-optimal trajectories in pink, blue and yellow. Part of the pink is obstructed by the blue and green path. The optimal strategy for O is $\lambda^* = [p(\mathbf{x}_1), p(\mathbf{x}_2), p(\mathbf{x}_3)] = [0.34, 0.32, 0.34]$, and the optimal strategy for E consists of three trajectories used with probability $\omega^* = \big[p(\text{blue}), p(\text{yellow}), p(\text{pink})\big] = [0.40, 0.20, 0.40]$. In this example, the pink and yellow λ^*-optimal trajectories initially coincide near $\mathbf{x}_S$, hence one cannot find both of them by perturbing the initial position $\mathbf{x}_S$.

among six possible positions. Using algorithm 6, we determine that at the approximate Nash equilibrium, only four positions are used with positive probability by O, and E uses four different trajectories which are displayed in fig. 4.10.

In order to test the performance of algorithm 6, we consider three error metrics:

1. *The optimization error* in $G(\lambda)$ arises from several effects: the discretization error of the Eikonal solver, the discretization error of the path tracing and path integral evaluation, and the early stopping of the supergradient iterations. To generate the "ground truth", we performed the same computation on a finer grid of size of $n = 2001 \times 2001$ (i.e. we consider a grid

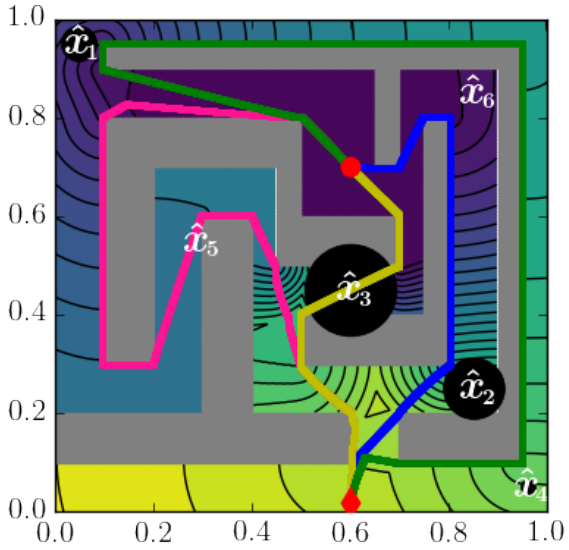


Figure 4.10: Computed Nash equilibrium for a maze-like example. The value function u^{λ^*} and four near-λ^*-optimal trajectories in pink, blue, yellow and green. The approximate Nash equilibrium strategy for O is $\lambda^* = [p(\hat{x}_i)]_{i=1}^{i=6} = [0.174, 0.301, 0.452, 0.073, 0, 0]$. The approximate Nash equilibrium strategy for E uses four trajectories with probability $\omega^* = \Big[p(\text{pink}), p(\text{yellow}), p(\text{blue}), p(\text{green})\Big] = [0.246, 0.461, 0.144, 0.149]$.

with 16 times more unknowns) and run the supergradient iteration until we observe stagnation in the objective function value of the iterates. We approximate the relative error in our computations on a 501×501 grid as:

$$E_{rel}\left[G(\lambda^*)\right] = \left|G_{501}(\lambda^*_{501}) - G_{2001}(\lambda^*_{2001})\right| \,/\, G_{501}(\lambda^*_{501}) \,.$$

2. *The Observer's regret* estimates how much the observer could improve his payoff by unilaterally deviating from our approximate Nash equilibrium. (Recall that, if the approximate Nash equilibrium were exact, the observer would not be able to increase his payoff at all). We quantify this error using the normalized residual in eq. (4.15), i.e.:

$$\text{Observer's regret} = \|R(\omega)\|_2 \;/\; (|\mathcal{I}|G(\lambda^*)) \;.$$

3. *The Evader's regret* estimates how much the evader could improve his pay-

off by unilaterally deviating from our approximate Nash equilibrium. This corresponds to how far from $\boldsymbol{\lambda}^*$-optimal are the controls produced by algorithm 6. Recall that the control function $\boldsymbol{a}_1(\cdot)$ is (up to numerical errors) $\boldsymbol{\lambda}^*$-optimal, whereas $\boldsymbol{a}_k(\cdot)$ for $k \geq 2$ are $(\boldsymbol{\lambda}^* + \delta\boldsymbol{\lambda})$-optimal. We report the maximum relative error in $\boldsymbol{\lambda}^*$ cumulative observability of the $(\boldsymbol{\lambda}^* + \delta\boldsymbol{\lambda})$-optimal trajectories, that is:

$$\text{Evader's regret} = \max_k \left| \mathcal{J}^{\boldsymbol{\lambda}^*}(\boldsymbol{a}_1(\cdot)) - \mathcal{J}^{\boldsymbol{\lambda}^*}(\boldsymbol{a}_k(\cdot)) \right| / \mathcal{J}^{\boldsymbol{\lambda}^*}(\boldsymbol{a}_1(\cdot))$$

These error metrics are reported in table 4.1 along with timing metrics for each example presented in the paper.

Table 4.1: Table of timing and error metrics. The error metrics are described in the main body of the text.

	fig. 4.3	fig. 4.5	fig. 4.6	fig. 4.9	fig. 4.10
Number of it. of algorithm 5	100	100	100	300	400
Total CPU time (seconds)	61	61	69	198	321
$E_{rel}\,[G(\boldsymbol{\lambda}^*)]$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$9 \cdot 10^{-4}$	$1 \cdot 10^{-3}$	$3 \cdot 10^{-4}$
Observer's regret	$1 \cdot 10^{-4}$	0	$3 \cdot 10^{-4}$	$4 \cdot 10^{-6}$	$1 \cdot 10^{-4}$
Evader's regret	0	0	$2 \cdot 10^{-3}$	$2 \cdot 10^{-3}$	$2 \cdot 10^{-2}$

4.6 Extension to groups of evaders

We now consider an extension of the surveillance-evasion game to a game which involves a team of q evaders. Each evader E^l chooses a trajectory leading him from his own source location $\mathbf{x}_S^l$ to a target location $\mathbf{x}_T^l$, according to his own speed function $f^l(x)$. The pointwise observability function $K^{\boldsymbol{\lambda}}$ is shared for all evaders and depends only on the strategy $\boldsymbol{\lambda}$ of the observer. This induces q

different cumulative observability functions $\mathcal{J}^{l,\lambda}(\mathbf{x}_S^l, \boldsymbol{a}^l(\cdot))$ defined as in eq. (4.1), and q different value functions $u^{l,\lambda}$ which are solutions of Eikonal equations with q different boundary conditions.

In this version of the game, we assume that a central organizer for evaders faces off against the observer. The goal of that central organizer is to minimize the weighted sum of evaders' cumulative expected observabilities. The weights $\{w_l\}_{l=1}^{l=q}$ in the sum reflect the relative importance of each evader. We further assume that the central organizer and the observer agree on that relative importance, making this a two player zero-sum game with a payoff function defined by:

$$P(\boldsymbol{\lambda}, \{\boldsymbol{\theta}^l\}_{l=1}^q) = \sum_{l=1}^{l=q} w_l \mathbb{E}_{\boldsymbol{\theta}^l}\left[\mathcal{J}^{l,\lambda}(\mathbf{x}_S^l, \boldsymbol{a}^l(\cdot))\right] . \tag{4.17}$$

Although we focus on a zero-sum two player game, we note that its Nash equilibrium $\left(\boldsymbol{\lambda}^*, \{\boldsymbol{\theta}^l\}_{l=1}^{l=q}\right)$ must also be among Nash equilibria of a different $(q+1)$-player game: the one, where each of the q evaders is selfishly minimizing their own cumulative observability $\mathcal{J}^{l,\lambda}(\mathbf{x}_S^l, \boldsymbol{a}_l(\cdot))$, while the observer still attempts to maximize the crowd-wide observability in eq. (4.17). This property follows from two simple facts:

1. The Observer's payoff is the same in both versions of the game and thus cannot be improved unilaterally in a $(q+1)$ player game.
2. In the Nash equilibrium for the two-player game, the central organizer would only ask each evader to assign positive probabilities to their $\boldsymbol{\lambda}^*$-optimal trajectories. (Otherwise, the weighted sum in eq. (4.17) could be improved). Thus, they would also be maximizing their individual payoffs.

In this new setting, theorem 4.4.2 holds and the observer's half of the Nash

equilibrium may be found by maximizing the concave function:

$$G^q(\boldsymbol{\lambda}) = \min_{\boldsymbol{a}^l(\cdot)} \sum_{l=1}^{l=q} w_l \mathcal{J}^{\boldsymbol{\lambda}}(\mathbf{x}_S^l, \boldsymbol{a}^l(\cdot)) \,. \tag{4.18}$$

The function $G^q(\boldsymbol{\lambda})$ and its supergradients may be evaluated in a similar way to section 4.5.2, but require q solves of the Eikonal equation with different boundary conditions and speed functions, and the numerical evaluation of $q \times r$ path integrals. However, we note that if all evaders have the same speed function and share the same target location (or, alternatively, share the same source location), only a single Eikonal equation solve is in fact required. With minor modifications, algorithm 6 may be also applied to solve this version of the problem. For each perturbation of $\boldsymbol{\lambda}^*$ a set of q control functions is generated on line 8 of algorithm 6, with one control function found for each evader. Although we obtain a new set of q control functions for each perturbation, some of the control functions for specific evaders may be essentially the same as those already obtained from previous perturbations. We address this in post-processing, by pruning the output of modified algorithm 6 to identify distinct trajectories for each evader.

We show numerical results for two test problems with $q = 2$ equally important evaders (i.e., $w_1 = w_2$) in each of them. An example presented in fig. 4.11 uses the same obstacle and the same $r = 2$ possible observer locations already used in fig. 4.6. At the approximate Nash equilibrium found using algorithm 6, the observer uses these two locations with probabilities $\boldsymbol{\lambda}^* = (0.35, 0.65)$ and the central controller directs both evaders to use pure policies: deterministically choose pink and blue trajectories to their respective targets. Even though the first evader's starting position and destination are also the same as in fig. 4.6, his (and the Observer's) optimal strategies are quite different here due to the

second evader's participation.

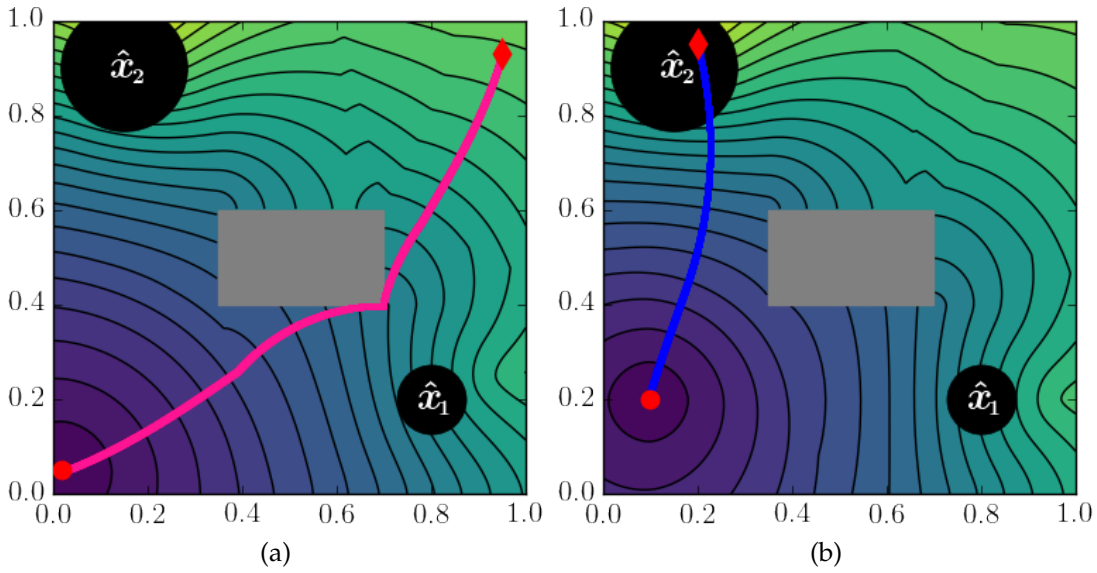


Figure 4.11: Computed approximate Nash equilibrium for a group of two evaders. The approximate Nash equilibrium pair of strategies is λ^* for O, and a single λ^*-optimal trajectory for each evader. (a) The value function u^{1,λ^*} for $\lambda^* = [0.35, 0.65]$ of evader 1, and the λ^*-optimal trajectories for evader 1 shown in pink. (b) The value function u^{2,λ^*} for the same λ^* of evader 2, and his λ^*-optimal trajectory shown in blue.

In a maze-like example presented in fig. 4.12, O can choose among six possible locations, but his optimal mixed strategy λ^* uses only four of them. algorithm 6 yields three sets of two near-λ^*-optimal trajectories which form an approximate Nash equilibrium, but they only contain two distinct trajectories for each of the evaders. We report timing and error metrics for these two examples in table 4.2.

Table 4.2: Table of running times and errors for examples with multiple evaders.

	fig. 4.11	fig. 4.12
Number of it. of algorithm 5	353	300
Total CPU time (seconds)	631	594
$E_{rel}\,[G(\lambda^*)]$	$5 \cdot 10^{-4}$	$7 \cdot 10^{-3}$
Observer's regret	$5 \cdot 10^{-4}$	$1 \cdot 10^{-3}$
Evader's regret	$5 \cdot 10^{-3}$	$1 \cdot 10^{-2}$

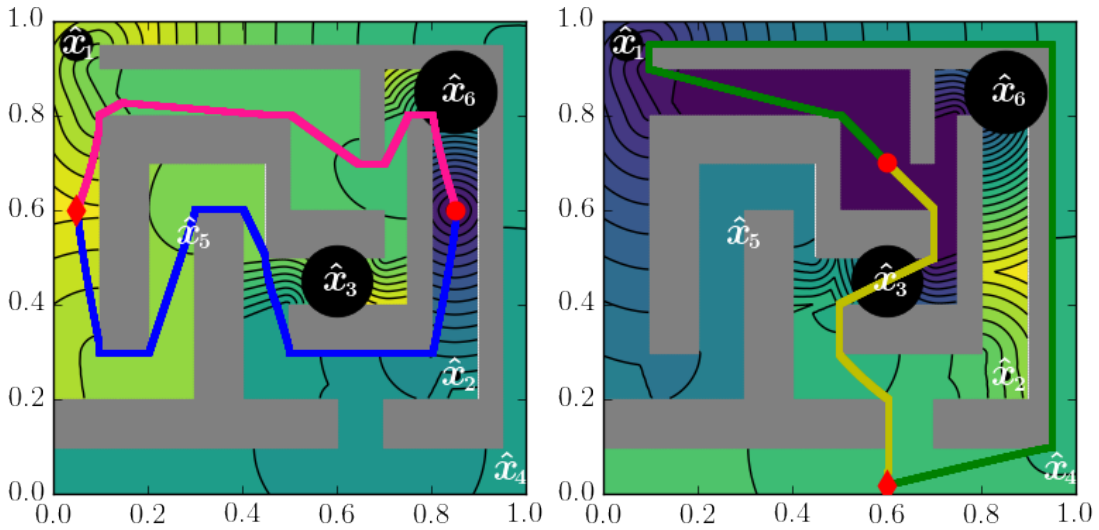


Figure 4.12: Computed approximate Nash equilibrium for a maze-like example with two evaders. (a) The value function u^{1,λ^*} of evader 1, and two near-λ^*-optimal trajectories for evader plotted in pink and blue. (b) The value function u^{2,λ^*} of evader 2, and two near-λ^*-optimal trajectories for evader 2 plotted in yellow and green. The approximate Nash equilibrium (λ^*, θ^*) is $\lambda^* = [p(\hat{x}_i)] = [0.168, 0.0455, 0.364, 0, 0, 0.422]$, and θ^* consists of a mixed strategy for the group of evaders. The mixed strategy of evader 1 is $\big[p(\text{pink}), p(\text{blue})\big] = [0.85, 0.15]$, and the mixed strategy for evader 2 is $\big[p(\text{yellow}), p(\text{green})\big] = [0.89, 0.11]$.

4.7 Conclusion

We have considered an adversarial path planning problem, where the goal is to minimize the cumulative exposure/observability to a hostile observer. The current position of the latter is unknown, but the full list of possible positions is assumed to be available in advance. The key assumption of our model is that neither the Evader (E) nor the enemy Observer (O) can adjust their plan in real time based on the opponent's state and actions. Instead, both of them are required to choose their (possibly randomized) strategies in advance. We discussed two versions of this problem; in the first one, a completely risk-averse evader attempts to minimize his worst-case cumulative observability. We showed that this version can be solved using previously developed methods for multiob-

jective path planning. However, the solution is prohibitively computationally expensive when O has a large number of surveillance plans to choose from. In the second version, the subject of optimization is the E's expected cumulative observability on its way to the target. We modeled this as a zero-sum Surveillance-Evasion Game (SEG) between two players: E (the minimizer) and O (the maximizer). We then presented an algorithm combining ideas from continuous optimal control, the scalarization approach for multiobjective optimization, and convex optimization which allows us to quickly compute an approximate Nash equilibrium of this semi-infinite strategic game. Finally, we showed that this algorithm extends to solve a similar problem involving a group of multiple evaders controlled by a central planner. The presented algorithm displays at most linear scaling in the number of observation plans, but further speed up techniques would be desirable; the computational bottleneck (numerically solving the Eikonal equation) could be alleviated with domain restriction methods [21] and factoring approaches [129].

Although this paper focused on isotropic problems, the anisotropic observer case could be treated in a similar fashion. (In practice, the pointwise observability might depend on the angle between the evader's direction of motion and the observer's line of sight.) This generalization will have to rely on fast numerical methods developed for anisotropic HJB PDEs; e.g., [2, 112, 146, 169]. In a follow-up paper [15], we show that time-dependent observation plans (e.g., different patrol routes) can be similarly treated by solving λ-parametrized finite-horizon optimal control problems with numerical methods for time-dependent HJB equations; e.g., [45, 151].

We note that the computational cost of our algorithm increases quickly with

the number of evaders considered. The case involving a large number of selfish evaders could be covered by considering the evolution of a time-dependent density of observers, and treating the problem using mean field games [14, 60]. Another possible extension would be to consider a group of observers choosing among a larger set of surveillance plans. In that situation, the set of pure strategies of the observers could increase exponentially, but we anticipate that the computational cost will grow much slower since the number of required Eikonal solves would not increase.

CHAPTER 5

CONTINUOUS ANALOGUES OF KRYLOV-SUBSPACE METHODS FOR DIFFERENTIAL OPERATORS

5.1 Introduction

Krylov subspace methods, such as the conjugate gradient (CG) method, MINRES, and GMRES, are iterative algorithms that solve $Ax = b$ using matrix-vector products [171]. After k iterations, they typically compute an approximate solution to $Ax = b$ from the Krylov subspace $\mathcal{K}_k(A, b) = \text{Span}\{b, Ab, \ldots, A^{k-1}b\}$. They provide a toolkit for solving large sparse or structured linear systems, which are omnipresent in computational mathematics. Krylov subspace methods are particularly prevalent in the context of solving differential equations; where a differential equation $\mathcal{L}u = f$ associated with a set of boundary conditions is discretized into a typically large sparse linear system $Ax = b$, and a Krylov subspace method is used to solve the resulting linear system.

During a discussion at the Chebfun and Beyond conference in September 2012 attended by about 150 numerical analysts,[1] the question was raised: can we design operator Krylov methods to solve differential equations without the need for discretization? In other words, can we build a Krylov-like method

This chapter is based on the paper "Continuous analogues of Krylov methods for differential operators" by M.A. Gilles and A. Townsend to appear in *SIAM Journal of Numerical Analysis*. The software based on this chapter is part of the Chebfun package, and is available at `http://www.chebfun.org/`.

[1]The session was chaired by Nick Higham. Alex Townsend scribed the discussion, following Nick Trefethen's advice.

for solving differential equations by building up a Krylov subspace through operator-function products? This is of particular interest for the spectral community as spectral discretizations are often dense, ill-conditioned, and do not always inherit the structure of the continuous problem (e.g., spectral discretizations of self-adjoint operators and not necessarily symmetric).

For these reasons, Krylov methods are not ubiquitously employed in the spectral method community [46, 165], despite $n \times n$ Chebyshev-based spectral discretization matrices of eq. (5.1) having a fast $O(n \log n)$ matrix-vector product based on the FFT [109]. In this paper, we describe the first practical implementation of operator analogues of Krylov methods for solving two-point boundary value problems (BVPs) [44, Chap. 6]. Although the presented method do not directly extend to partial differential equations, we discuss possible ways forward in section 5.6. In order to simplify the exposition, we proceed with the assumption that $\mathcal{L}$ is a second-order operator, but we extend the methods to all even-ordered ODEs in section 5.5. Thus, we focus on a simple problem of the form:

$$\mathcal{L}u = f \quad \text{on } \Omega = (-1, 1), \qquad u(\pm 1) = 0, \tag{5.1}$$

where $\mathcal{L}u = -(a(x)u'(x))' + b(x)u'(x) + c(x)u$, $\mathcal{L} : \mathcal{H}_0^1(\Omega) \cap \mathcal{H}^2(\Omega) \to L^2(\Omega)$[2], $a, b, c \in L^\infty(\Omega)$ and $f \in L^2(\Omega)$.

If there are no additional assumptions on $\mathcal{L}$, then we propose an analogue of GMRES to solve eq. (5.1) (see section 5.4.1). If $b(x) = 0$, then $\mathcal{L}$ is self-adjoint, which is analogous to a symmetric matrix, and we propose an analogue of MINRES (see section 5.4.2). When $a(x) > 0$, $b(x) = 0$, and $c(x) \geq 0$, $\mathcal{L}$ is self-adjoint

[2] This is a slight restriction from the more typical $\mathcal{L} : \mathcal{H}_0^1(\Omega) \to \mathcal{H}^{-1}(\Omega)$ setup. However, this restriction is natural in the present context where we develop practical algorithms which apply $\mathcal{L}$ to functions by weak differentiation operations and function products instead of having to revert to a bilinear form interpretation of the function product (see section 5.3).

with real positive eigenvalues [44, Sec. 6.5, Thm. 1], which is analogous to a symmetric positive definite matrix, and we propose an analogue of the CG method (see section 5.2).

To see the difficulties in developing a Krylov-based method for differential operators, consider solving $-u''(x) = 1 - x^2$ on $\Omega = (-1, 1)$ with $u(\pm 1) = 0$. The exact solution is $u(x) = (x^4 - 6x^2 + 5)/12$. A naive generalization of the Krylov subspace is $\mathcal{K}_k(\mathcal{L}, f) = \text{Span}\{f, \mathcal{L}f, \mathcal{L}(\mathcal{L}f), \ldots, \mathcal{L}^{k-1}f\}$ with $f = 1 - x^2$. Since $\mathcal{L}u = -u''$, this leads to $\mathcal{K}_k(\mathcal{L}, f) = \text{Span}\{1 - x^2, 2\}$ for $k \geq 2$. This example illustrates that such an approach is flawed, as $\mathcal{K}_k(\mathcal{L}, f)$ does not contain a good approximation to the exact solution. Moreover, the boundary conditions are not imposed because $\mathcal{K}_k(\mathcal{L}, f) \not\subset \mathcal{H}_0^1(\Omega)$ for $k \geq 2$. There are at least three major theoretical issues to overcome:

Problem 5.1.1. *Since $f \notin \mathcal{H}_0^1(\Omega)$ and $\text{Range}(\mathcal{L}) \not\subset \mathcal{H}_0^1(\Omega)$, how does one construct a Krylov subspace that satisfies the boundary conditions? Our answer involves using orthogonal projection operators to ensure that each term in the Krylov subspace is in $\mathcal{H}_0^1(\Omega)$ (see section 5.2.1), and solving an ancillary problem (see section 5.2.5).*

Problem 5.1.2. *Since $\mathcal{L} : \mathcal{H}_0^1(\Omega) \cap \mathcal{H}^2(\Omega) \to L^2(\Omega)$, how does one repeatedly apply operator-function products that are necessary to build up a Krylov subspace? To achieve this, we use an orthogonal projection operator and a preconditioner that acts as a "smoother" (see section 5.2.2).*

Problem 5.1.3. *Since $\mathcal{L}$ is an unbounded operator, how does one construct a Krylov method that rapidly converges to the solution of eq.* (5.1)*? Our answer is to use operator preconditioners that allow for our Krylov iterations to be terminated after a finite number of iterations with an approximate solution (see section 5.2.4).*

The Krylov methods that we develop solve eq. (5.1) by directly applying $\mathcal{L}$ to functions, and we prove that the iterates from our preconditioned CG method geometrically converge to the solution (see corollary 5.2.2). Our operator Krylov methods are not equivalent to matrix Krylov methods applied to a standard discretization of eq. (5.1), and offer several advantages: (1) Operator preconditioners are motivated by the differential operator as opposed to the properties of a discretization scheme, (2) The resulting CG method can always be applied to eq. (5.1) with $a(x) > 0$, $b(x) = 0$, and $c(x) \geq 0$ without the need for structure-preserving discretizations [149, Chap. 4], (3) The iterates converge to the desired solution of eq. (5.1), as opposed to the solution of a discretization, and (4) The method is fully adaptive: it automatically chooses the complexity needed to represent each of the iterates.

Several attempts to develop operator Krylov methods for differential equations have been proposed that we believe date back to 1967 [27], where it was shown that an operator CG method can be reduced to a sequence of Poisson problems with Dirichlet boundary conditions. In 2009, a promising differential GMRES method for computing oscillatory integrals [117] was developed in the context of spectral methods, but it has remained unclear how to successfully incorporate boundary conditions. A theoretical foundation for a CG method on ordinary and partial differential operators [106] was introduced in 2015. The

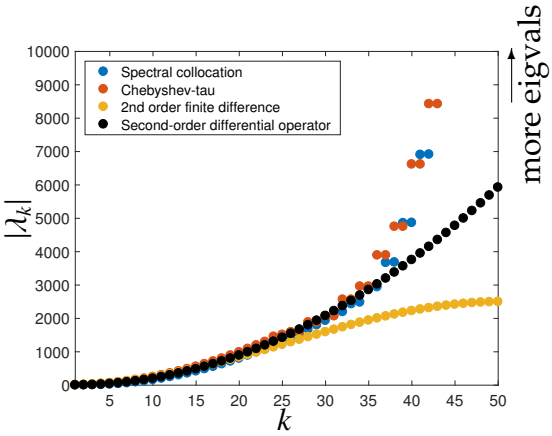


Figure 5.1: Spectra of 50×50 discretizations of $\mathcal{L}u = -u''$ with zero Dirichlet boundary conditions. Similar figures appear in [176, Fig. 1]. Spectral collocation (blue dots) [165], and Chebyshev tau (red dots) [119] discretizations typically have spectra that grow asymptotically faster than the spectra of the underlying differential operator (black dots), while the spectra of finite difference (yellow dots) [92] discretizations grow asymptotically slower. Most popular spectral discretizations are more ill-conditioned than expected, leading to poor convergence properties of Krylov subspace solvers. Our operator Krylov methods avoid discretizing BVPs and employs preconditioners that are motivated from the differential operator (see section 5.2.2).

authors use a Riesz map $\tau : \mathcal{H}^{-1}(\Omega) \to \mathcal{H}_0^1(\Omega)$ to precondition a differential operator [106, Chap. 4] and successfully construct a Krylov subspace of the form $\text{Span}\{\tau f, \tau\mathcal{L}\tau f, (\tau\mathcal{L})^2\tau f, \dots, \}$. This work is an insightful theoretical framework and our paper expands on their contribution in order to develop a collection of practical Krylov methods for solving eq. (5.1).

Though we do not discretize the differential operator itself, for our operator Krylov methods to be of practical interest, one must employ an approximation space for the solution and right-hand side (see section 5.3). Unlike most BVP solvers, the approximation space can be all of $\mathcal{H}_0^1(\Omega)$ or an infinite dimensional dense subspace of $\mathcal{H}_0^1(\Omega)$. This allows one to implement highly adaptive Krylov subspace methods that automatically resolve the solution to machine precision (see section 5.3).

Intuitively, our main idea is to modify the operator-function products with $\mathcal{L}$ while preserving the weak form of eq. (5.1). That is, we respect the bilinear form [44, p. 316] associated with eq. (5.1), i.e.,

$$\mathcal{B}[\phi,\psi] = \int_{-1}^{1} a(x)\phi'(x)\psi'(x) + b(x)\phi'(x)\psi(x) + c(x)\phi(x)\psi(x)dx, \qquad \phi,\psi \in \mathcal{H}_0^1(\Omega) \tag{5.2}$$

as well as the weak form of the solution as $\mathcal{B}[u,\psi] = \langle f,\psi\rangle$ for all $\psi \in \mathcal{H}_0^1(\Omega)$. Here, and throughout the paper, we use $\langle\cdot,\cdot\rangle$ to denote the standard L^2 inner-product and $\|\psi\|^2 = \langle\psi,\psi\rangle$.

The paper is structured as follows. In section 5.2 we derive an unpreconditioned and preconditioned CG method for solving eq. (5.1) when $\mathcal{L}$ is a self-adjoint second-order differential operator with $a(x) > 0$, $b(x) = 0$, and $c(x) \geq 0$. In section 5.3 we use our CG theory to develop practical iterative BVP solvers for eq. (5.1). In section 5.4, we extend our CG method to operator analogues of MINRES and GMRES. In section 5.5 we show how our ideas can be applied to higher-order BVPs, and in section 5.6 we tentatively consider PDEs.

5.2 The CG method for differential operators

The CG method for matrices is an iterative algorithm for solving $Ax = b$, where A is a symmetric positive definite matrix [71]. It constructs iterates $x_0 = 0, x_1, x_2, \ldots$, such that x_k is the best approximate from $\mathcal{K}_k(A,b)$ as measured by the energy norm. That is,

$$x_k = \underset{y\in K_k(A,b)}{\arg\min} \|x-y\|_A, \qquad K_k(A,b) = \text{Span}\left\{b, Ab, \ldots, A^{k-1}b\right\},$$

where $\|y\|_A^2 = y^TAy$ and $x = A^{-1}b$ is the exact solution. The fact that $\|\cdot\|_A$ defines a norm is central to the development and analysis of the CG method for

matrices [98, Sec. 5.6].

Just like symmetric positive definite matrices, self-adjoint differential operators with $a(x) > 0$ and $c(x) \geq 0$ have real positive eigenvalues and an orthogonal basis of eigenfunctions [44, Sec. 6.5, Thm. 1]. The analogue of the energy norm in this setting is $\|\phi\|_{\mathcal{L}}^2 = \mathcal{B}[\phi, \phi]$ for $\phi \in \mathcal{H}_0^1(\Omega)$, where $\mathcal{B}$ is the bilinear form associated to $\mathcal{L}$ in eq. (5.2). The fact that $\|\cdot\|_{\mathcal{L}}$ defines a norm is equally important for the development and analysis of a CG method for eq. (5.1).

If $p_0, p_1, \ldots,$ form a complete basis for $\mathcal{H}_0^1(\Omega)$ so that $\mathcal{B}[p_i, p_j] = 0$ for $i \neq j \geq 0$, then since $f \in L^2(\Omega)$, we may formally write the solution to eq. (5.1) as

$$u = \sum_{j=0}^{\infty} \frac{\left\langle f, p_j \right\rangle}{\mathcal{B}[p_j, p_j]} p_j.$$

Our CG method carefully constructs functions $p_0, p_1, \ldots,$ sequentially, such that $\mathcal{B}[p_i, p_j] = 0$ for $i \neq j$, in the hope that we may not need all of them to obtain a good approximation to u.

5.2.1 The unpreconditioned CG method with a restricted right-hand side

In order to tackle the first major issue highlighted in the introduction (see Problem 5.1.1), we compose $\mathcal{L}$ with a projection operator[3] to ensure that any solution from the constructed Krylov subspace satisfies the zero Dirichlet conditions of eq. (5.1).

Let $\mathcal{V}_0$ be an approximation space for the solution of eq. (5.1). We wish to

[3]The idea of composing a matrix with a projection operator to generate a Krylov subspace is also used for solving saddle-point problems [59].

construct a projection onto $\mathcal{V}_0$ and apply it after each operator-function product so that the constructed Krylov subspace is a subspace of $\mathcal{V}_0$. We temporarily make the following assumptions:

Assumption 1. *$\mathcal{V}_0$ is a closed (potentially infinite-dimensional) subspace of the solution space $\mathcal{H}_0^1(\Omega) \cap \mathcal{H}^2(\Omega)$, and*

Assumption 2. $f \in \mathcal{V}_0$.

In section 5.2.2, we introduce a preconditioner that acts as a "smoother" to eliminate the need for Assumption 1 and allows us to set $\mathcal{V}_0 = \mathcal{H}_0^1(\Omega)$. We avoid Assumption 2 by solving an ancillary problem (see section 5.2.5).

Proceeding under Assumptions 1 and 2, we define an orthogonal projection operator onto $\mathcal{V}_0$ (because $\mathcal{V}_0$ is a closed subspace of $L^2(\Omega)$) as

$$\Pi_{\mathcal{V}_0}\phi = \underset{p\in\mathcal{V}_0}{\arg\min} \|\phi - p\|, \qquad \Pi_{\mathcal{V}_0} : L^2(\Omega) \to \mathcal{V}_0.$$

We work with the modified operator $\Pi^*_{\mathcal{V}_0}\mathcal{L}\Pi_{\mathcal{V}_0} : L^2(\Omega) \to \mathcal{V}_0$, where $\Pi^*_{\mathcal{V}_0} : L^2(\Omega) \to \mathcal{V}_0$ is the adjoint of $\Pi_{\mathcal{V}_0}$ over the L^2 inner-product. Since $\Pi_{\mathcal{V}_0}$ is an orthogonal projection, it is self-adjoint, i.e., $\Pi^*_{\mathcal{V}_0} = \Pi_{\mathcal{V}_0}$ [135, Chap. 5]. This is important as it implies that the range of $\Pi^*_{\mathcal{V}_0}\mathcal{L}\Pi_{\mathcal{V}_0}$ is $\mathcal{V}_0$, and that the operator $\Pi^*_{\mathcal{V}_0}\mathcal{L}\Pi_{\mathcal{V}_0}$ is self-adjoint. Consequently, it is reasonable to imagine applying a CG method with $\Pi^*_{\mathcal{V}_0}\mathcal{L}\Pi_{\mathcal{V}_0}$.

The operator $\Pi^*_{\mathcal{V}_0}\mathcal{L}\Pi_{\mathcal{V}_0} : L^2(\Omega) \to \mathcal{V}_0$ is well-defined since $\mathcal{V}_0 \subset \mathcal{H}_0^1(\Omega) \cap \mathcal{H}^2(\Omega)$, and we are interested in Krylov subspaces of the form

$$\mathcal{K}_k(\Pi^*_{\mathcal{V}_0}\mathcal{L}\Pi_{\mathcal{V}_0}, f) = \mathrm{Span}\left\{f, \Pi^*_{\mathcal{V}_0}\mathcal{L}\Pi_{\mathcal{V}_0} f, \ldots, (\Pi^*_{\mathcal{V}_0}\mathcal{L}\Pi_{\mathcal{V}_0})^{k-1} f\right\}, \qquad k \geq 1. \tag{5.3}$$

Since $f \in \mathcal{V}_0$, we know that $\mathcal{K}_k(\Pi^*_{\mathcal{V}_0}\mathcal{L}\Pi_{\mathcal{V}_0}, f) \subseteq \mathcal{V}_0$ so that the boundary conditions are successfully incorporated into the Krylov subspace. Therefore, any iterative method that constructs iterates from the Krylov subspace in eq. (5.3) automatically imposes zero Dirichlet boundary conditions.

An unpreconditioned CG method can now be derived that generates iterates $u_0 = 0, u_1, u_2, \ldots$, such that

$$u_k = \underset{v \in \mathcal{K}_k(\Pi^*_{\mathcal{V}_0}\mathcal{L}\Pi_{\mathcal{V}_0}, f)}{\arg\min} \|u - v\|_{\mathcal{L}},$$

where u is the exact solution to eq. (5.1)[4]. The derivation of this method follows almost immediately from the CG method for matrices [167, Alg. 38.1], where in the derivation terms of the form $x^T A y$ are replaced by $\mathcal{B}[\phi, \psi]$, $x^T y$ by $\langle \phi, \psi \rangle$, and Ax by $\Pi^*_{\mathcal{V}_0}\mathcal{L}\Pi_{\mathcal{V}_0}\phi$. The resulting unpreconditioned CG method for eq. (5.1) is given in algorithm 9. We also give the matrix CG method in algorithm 8 for comparison, and we emphasize that the two algorithms are essentially the same except the operations algorithm 8 are with vectors and matrices while the operations in algorithm 9 are with functions and operators.

[4] This follows from the fact that the discretization error is $\mathcal{B}$-orthogonal to the algebraic error in a Galerkin method [98, Thm. 2.5.2].

Algorithm 8 The CG method for solving $Ax = b$, where $A \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix and $b \in \mathbb{R}^{n\times 1}$.

Set $x_0 = 0$, $r_0 = b$, and $p_0 = b$
for $k = 0, 1, \ldots$ (until converged) **do**
  $\alpha_k = r_k^T r_k / (p_k^T A p_k)$
  $x_{k+1} = x_k + \alpha_k p_k$
  $r_{k+1} = r_k - \alpha_k A p_k$
  $\beta_k = r_{k+1}^T r_{k+1} / r_k^T r_k$
  $p_{k+1} = r_{k+1} + \beta_k p_k$
end for

Algorithm 9 The CG method for eq. (5.1), where $\mathcal{L}$ is self-adjoint with $a(x) > 0$ and $c(x) \geq 0$, and $f \in \mathcal{V}_0$.

Set $u_0 = 0$, $r_0 = f$, and $p_0 = f$
for $k = 0, 1, \ldots$ (until converged) **do**
  $\alpha_k = \langle r_k, r_k \rangle / \mathcal{B}[p_k, p_k]$
  $u_{k+1} = u_k + \alpha_k p_k$
  $r_{k+1} = r_k - \alpha_k \Pi_{\mathcal{V}_0}^* \mathcal{L} \Pi_{\mathcal{V}_0} p_k$
  $\beta_k = \langle r_{k+1}, r_{k+1} \rangle / \langle r_k, r_k \rangle$
  $p_{k+1} = r_{k+1} + \beta_k p_k$
end for

For algorithm 9 to be well-defined we must check that: (1) $r_0, r_1, \ldots$, are in $L^2(\Omega)$ so that $\langle r_k, r_k \rangle$ is valid, (2) $p_0, p_1, \ldots$, are in $L^2(\Omega)$ so that $\Pi_{\mathcal{V}_0}^* \mathcal{L} \Pi_{\mathcal{V}_0} p_k$ is well-defined, and (3) $p_0, p_1, \ldots$, are in $\mathcal{H}_0^1(\Omega)$ so that $\mathcal{B}[p_k, p_k]$ is valid. All these statements hold when $f \in \mathcal{V}_0 \subset \mathcal{H}_0^1(\Omega) \cap \mathcal{H}^2(\Omega)$ and can be verified by mathematical induction.

The CG method in algorithm 9 is theoretically justified for any $\mathcal{V}_0$ that is a closed subspace of $\mathcal{H}_0^1(\Omega) \cap \mathcal{H}^2(\Omega)$. In particular, this includes the space $\mathcal{V}_0 = \{v \in \mathcal{P}_n : v(\pm 1) = 0\}$ for some integer n, where $\mathcal{P}_n$ is the space of polynomials of degree $\leq n$. Furthermore, if the basis for $\mathcal{P}_n$ is selected to be the Legendre polynomials, then the CG method in algorithm 9 is closely related to applying the CG method to a Legendre–Galerkin discretization of eq. (5.1) [149, Sec. 4.1]. The advantage of algorithm 9 is that it provides important insights into how to derive a preconditioned CG method (see section 5.2.3).

The convergence of the unpreconditioned CG method in algorithm 9 is generically poor (see fig. 5.2). The unboundedness of the differential operator

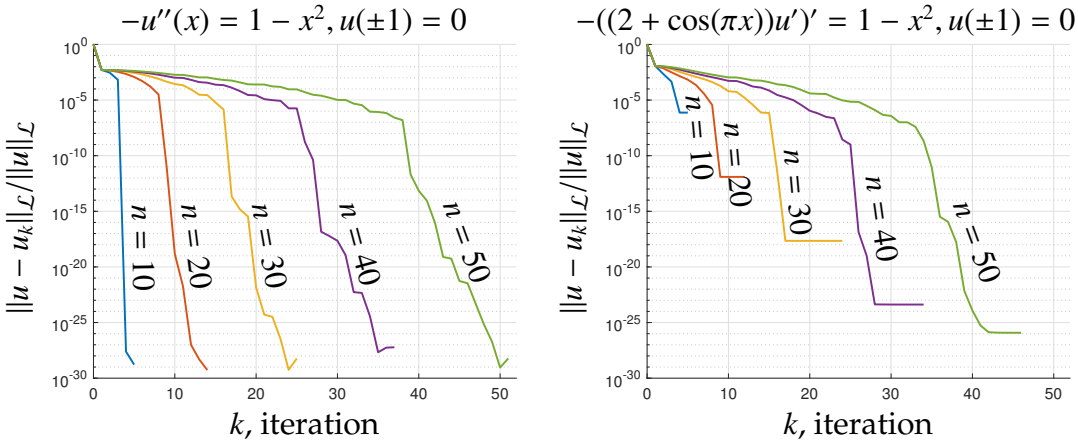


Figure 5.2: Convergence of the unpreconditioned CG method when $\mathcal{V}_0 = \{v \in \mathcal{P}_n : v(\pm 1) = 0\}$ and $10 \leq n \leq 50$, where $\mathcal{P}_n$ is the space of polynomials of degree $\leq n$. Left: The CG error when solving $-u'' = 1 - x^2$ on $(-1, 1)$ and $u(\pm 1) = 0$. Right: The CG error when solving $-((2 + \cos(\pi x))u')' = 1 - x^2$ on $(-1, 1)$ and $u(\pm 1) = 0$. The unpreconditioned CG method here is rarely useful because differential operators are unbounded and the number of required CG iterations is generically $\dim(\mathcal{V}_0)$. To overcome this, we develop operator preconditioners (see section 5.2.2).

means that $k = \dim(\mathcal{V}_0)$ iterations are typically necessary (see fig. 5.2) and, in our setting, $\mathcal{V}_0$ could potentially be an infinite dimensional subspace.

5.2.2 Operator preconditioning

Improving the convergence of algorithm 9 requires the development of preconditioners. The preconditioned CG method for solving $Ax = b$ is equivalent to applying the CG method to $R^T ARy = R^T b$, where $x = Ry$ and R is a square matrix [110, Sec. 8.1]. Motivated by this, we consider solving

$$\mathcal{R}^* \mathcal{L} \mathcal{R} v = \mathcal{R}^* f \quad \text{on } \Omega = (-1, 1), \qquad (\mathcal{R}v)(\pm 1) = 0, \tag{5.4}$$

where $\mathcal{R} : L^2(\Omega) \to L^2(\Omega)$ is a linear operator and $\mathcal{R}^*$ is the adjoint of $\mathcal{R}$, i.e., $\langle \mathcal{R}^* \phi, \psi \rangle = \langle \phi, \mathcal{R}\psi \rangle$ for all $\phi, \psi \in L^2(\Omega)$. We call $\mathcal{R}$ an *operator preconditioner*.[5]

[5] In the Petrov–Galerkin literature, the concept of "operator preconditioning" is similar and refers to a recipe for constructing preconditioners so that they are robust with respect to the

We make the following requirements on the operator preconditioner $\mathcal{R} : L^2(\Omega) \to L^2(\Omega)$, which appear to be necessary in our framework to overcome the remaining two major issues highlighted in the introduction (see Problems 5.1.2 and 5.1.3):

Bounded: The preconditioner $\mathcal{R} : L^2(\Omega) \to L^2(\Omega)$ is a bounded linear operator. That is, $\|\mathcal{R}\|_{\text{op}} = \sup_{\phi \in L^2(\Omega), \|\phi\|=1} \|\mathcal{R}\phi\| < \infty$.

Smoother: The preconditioner and its adjoint over $L^2(\Omega)$ are smoothers, i.e., $\mathcal{R} : L^2(\Omega) \to \mathcal{H}^1(\Omega)$, $\mathcal{R} : \mathcal{H}^1(\Omega) \to \mathcal{H}^2(\Omega)$, $\mathcal{R}^* : L^2(\Omega) \to \mathcal{H}^1(\Omega)$, and $\mathcal{R}^* : \mathcal{H}^1(\Omega) \to \mathcal{H}^2(\Omega)$.[6]

Preconditioner for the Laplacian: There are constants $0 < \gamma_0 \leq \gamma_1 < \infty$ such that $\gamma_0 \|\phi\|^2 \leq \|(\mathcal{R}\phi)'\|^2 \leq \gamma_1 \|\phi\|^2$ for all $\phi \in L^2(\Omega)$.

A natural operator preconditioner for eq. (5.1), and our canonical choice, is the indefinite integration operator $\mathcal{R} : L^2(\Omega) \to L^2(\Omega)$, defined as

$$(\mathcal{R}\phi)(x) = \int_{-1}^{x} \phi(s)ds, \qquad (\mathcal{R}^*\phi)(x) = \int_{x}^{1} \phi(s)ds, \qquad \phi \in L^2(\Omega). \tag{5.5}$$

The preconditioner and its adjoint act as "smoothers" and $\|\mathcal{R}\|_{\text{op}} = 4/\pi < \infty$ [68, Prob. 188]. If $\mathcal{L}u = -u''$, then the associated bilinear form of the operator $\mathcal{R}^*\mathcal{L}\mathcal{R}$ is

$$\mathcal{B}[\mathcal{R}\phi, \mathcal{R}\psi] = \int_{-1}^{1} \left(\int_{-1}^{x} \phi(s)ds \right)' \left(\int_{-1}^{x} \psi(s)ds \right)' dx = \int_{-1}^{1} \phi(x)\psi(x)dx = \langle \phi, \psi \rangle ,$$

where $\phi, \psi \in L^2(\Omega)$. Therefore, $\mathcal{R}$ is a preconditioner for the Laplacian with $\gamma_0 = \gamma_1 = 1$ so that the $\mathcal{R}$ in eq. (5.5) satisfies all of our requirements.

choice of trial and test basis [72]. A related concept is "equivalent preconditioners", where one constructs a preconditioner by simplifying the given differential operator [5].

[6] Note that if $f \in L^2(\Omega)$ this implies that $\mathcal{R}^* f \in \mathcal{H}^1(\Omega)$ and $\mathcal{R}^*\mathcal{L}\mathcal{R} : \mathcal{H}^1(\Omega) \to \mathcal{H}^1(\Omega)$.

The integral preconditioner in eq. (5.5) appears throughout the literature and is exploited to construct preconditioners for finite element discretizations [88] as well as for spectral Galerkin discretizations [13, Chap. 4].

5.2.3 The preconditioned CG method

With an operator preconditioner in hand, we are able to derive a satisfying operator CG method. In order for $\mathcal{H}_0^1(\Omega)$ to be the solution space for $u = \mathcal{R}v$ in eq. (5.4), the space $\mathcal{W}_0 = \{\phi \in L^2(\Omega) : \mathcal{R}\phi \in \mathcal{H}_0^1(\Omega)\}$ must be the approximation space for v in eq. (5.4). Moreover, instead of assuming that $f \in \mathcal{V}_0$, we must now work under the the following assumption:

Assumption 3. $\mathcal{R}^* f \in \mathcal{W}_0$.

We no longer need Assumption 1 and we remove Assumption 3 in section 5.2.5. Since we are using a preconditioner and the approximation space for the solution of eq. (5.4) is $\mathcal{W}_0$, we first need to design an orthogonal projection operator $\Pi_{\mathcal{W}_0} : L^2(\Omega) \to \mathcal{W}_0$. This task appears challenging for general preconditioners $\mathcal{R}$. However, when $\mathcal{R}$ is the indefinite integral preconditioner in eq. (5.5), we note that $\mathcal{W}_0$ is the space of $L^2(\Omega)$ functions with zero mean. Moreover, $(\mathcal{R}\phi)(-1) = 0$ for all $\phi \in L^2(\Omega)$, and hence we find that the orthogonal projection $\Pi_{\mathcal{W}_0} : L^2(\Omega) \to \mathcal{W}_0$ is given by

$$\Pi_{\mathcal{W}_0}\phi = \phi - \frac{1}{2}\int_{-1}^{1} \phi(s)ds.$$

It is easy to verify that this projection operator is self-adjoint, and thus orthogonal:

$$\langle \Pi_{\mathcal{W}_0}\phi, \psi\rangle = \langle \phi, \psi\rangle - \frac{1}{2}\int_{-1}^{1}\phi(s)ds\int_{-1}^{1}\psi(s)ds = \langle \phi, \Pi_{\mathcal{W}_0}\psi\rangle, \qquad \phi, \psi \in L^2(\Omega).$$

Given an orthogonal projection operator $\Pi_{\mathcal{W}_0} : L^2(\Omega) \to \mathcal{W}_0$, we can derive a preconditioned CG method that constructs iterates $v_0 = 0, v_1, v_2, \ldots$, so that $u_k = \mathcal{R}v_k$ approximates the solution to eq. (5.1). The Krylov subspace of interest is now

$$\mathcal{K}_k(\mathcal{T}, \mathcal{R}^* f) = \text{Span}\left\{\mathcal{R}^* f, \mathcal{T}(\mathcal{R}^* f), \ldots, \mathcal{T}^{k-1}\mathcal{R}^* f\right\}, \qquad \mathcal{T} = \Pi^*_{\mathcal{W}_0}\mathcal{R}^*\mathcal{L}\mathcal{R}\Pi_{\mathcal{W}_0}, \quad (5.6)$$

where $\mathcal{K}_k(\mathcal{T}, \mathcal{R}^* f) \subset \mathcal{W}_0$ because Assumption 3 ensures that $\mathcal{R}^* f \in \mathcal{W}_0$. The associated preconditioned CG method is given in algorithm 10.

Algorithm 10 The preconditioned CG method for eq. (5.1), where $\mathcal{L}$ is self-adjoint with $a(x) > 0$ and $c(x) \geq 0$, and $\mathcal{R}^* f \in \mathcal{W}_0$.

1: Set $v_0 = 0$, $r_0 = \mathcal{R}^* f$, and $p_0 = \mathcal{R}^* f$
2: **for** $k = 0, 1, \ldots,$ (until converged) **do**
3: $\quad \alpha_k = \langle r_k, r_k\rangle / \mathcal{B}[\mathcal{R}p_k, \mathcal{R}p_k]$
4: $\quad v_{k+1} = v_k + \alpha_k p_k$
5: $\quad r_{k+1} = r_k - \alpha_k \mathcal{T} p_k$
6: $\quad \beta_k = \langle r_{k+1}, r_{k+1}\rangle / \langle r_k, r_k\rangle$
7: $\quad p_{k+1} = r_{k+1} + \beta_k p_k$
8: $\quad u_{k+1} = \mathcal{R}v_{k+1}$
9: **end for**

To verify that algorithm 10 is well-defined we check that: (1) $r_0, r_1, \ldots,$ are in $L^2(\Omega)$ so that $\langle r_k, r_k\rangle$ is valid, (2) $p_0, p_1, \ldots,$ are in $L^2(\Omega)$ so that $\mathcal{T}p_k$ and $\mathcal{B}[\mathcal{R}p_k, \mathcal{R}p_k]$ are valid operations. All these statements hold when $\mathcal{R}^* f \in \mathcal{W}_0$ and can be proved by mathematical induction.

The preconditioned CG method in algorithm 10 immediately inherits many of the theoretical properties from the CG method for matrices [110]. Here are

two immediate facts that are analogous to familiar results for the matrix CG method:

Lemma 5.2.1. *The functions $r_0, r_1, \ldots,$ in algorithm 10 satisfy $\left\langle r_i, r_j \right\rangle = 0$ for $i \neq j$. Moreover, the functions $p_0, p_1, \ldots,$ satisfy $\mathcal{B}[\mathcal{R}p_i, \mathcal{R}p_j] = 0$ for $i \neq j$.*

Proof. The constant α_k is selected so that $\langle r_{k+1}, r_k \rangle = 0$ for $k \geq 0$. This gives the formula $\alpha_k = \langle r_k, r_k \rangle / \mathcal{B}[\mathcal{R}r_k, \mathcal{R}p_k]$, which can be simplified to the formula in algorithm 10 since $r_{k+1} = p_{k+1} - \beta_k p_k$. The constant β_k is selected so that $\mathcal{B}[\mathcal{R}p_{k+1}, \mathcal{R}p_k] = 0$ for $k \geq 0$. This gives the formula $\beta_k = -\mathcal{B}[\mathcal{R}r_{k+1}, \mathcal{R}p_k]/\mathcal{B}[\mathcal{R}p_k, \mathcal{R}p_k]$, which can be simplified to the formula in algorithm 10 since $r_{k+1} = r_k - \alpha_k \mathcal{T} p_k$. The result immediately follows. □

Lemma 5.2.1 also shows that algorithm 10 is solving a best approximation problem.

Theorem 5.2.1. *Let $u_0 = 0, u_1, \ldots,$ be the CG iterates from algorithm 10 and u the solution to eq.* (5.1)*. Then,*

$$u_k = \underset{p \in \mathcal{X}_k}{\arg\min} \|u - p\|_{\mathcal{L}}, \qquad k \geq 1,$$

where $\mathcal{X}_k = \{p \in \mathcal{H}_0^1(\Omega) : p = \mathcal{R}q, q \in K_k(\mathcal{T}, \mathcal{R}^* f)\}$.

Proof. From lemma 5.2.1, we find that $\mathcal{B}[\mathcal{R}(v - v_k), \mathcal{R}p_j] = 0$ for $j \geq k + 1$, where $u = \mathcal{R}v$. In other words, we have

$$v_k = \underset{q \in K_k(\mathcal{T}, \mathcal{R}^* f)}{\arg\min} \|v - q\|_{\mathcal{T}}.$$

Since $\|v-q\|_{\mathcal{T}}^2 = \mathcal{B}[\mathcal{R}(v-q), \mathcal{R}(v-q)] = \mathcal{B}[u-\mathcal{R}q, u-\mathcal{R}q] = \|u-\mathcal{R}q\|_{\mathcal{L}}^2$, this is equivalent to $v_k = \arg\min_{q \in K_k(\mathcal{T}, \mathcal{R}^* f)} \|u - \mathcal{R}q\|_{\mathcal{L}}$. Finally, we note that $u_k = \mathcal{R}v_k$ and therefore, $u_k = \arg\min_{p \in \mathcal{X}_k} \|u - p\|_{\mathcal{L}}$, where $\mathcal{X}_k = \{p \in \mathcal{H}_0^1(\Omega) : p = \mathcal{R}q, q \in K_k(\mathcal{T}, \mathcal{R}^* f)\}$. □

Theorem 5.2.1 shows that algorithm 10 is calculating the best approximation from $\mathcal{X}_k$ to u in the $\|\cdot\|_{\mathcal{L}}$ norm and also guarantees that the error $e_k = \|u - u_k\|_{\mathcal{L}}$ is monotonically non-increasing, i.e.,

$$\|u - u_{k+1}\|_{\mathcal{L}} \leq \|u - u_k\|_{\mathcal{L}}, \qquad k \geq 0.$$

In practice, designing good preconditioners is paramount for an efficient BVP solver. One could imagine being confronted with the same dilemma as preconditioning the CG method for matrices. On the one hand, we want to select $\mathcal{R}$ so that $\mathcal{R}\phi$ can be computed efficiently for any $\phi \in \mathcal{W}_0$. On the other hand, we want $\mathcal{T}$ to be a well-conditioned operator over $\mathcal{W}_0$ (see eq. (5.7)). Here, we have an additional desire that is not present for matrices: we would like an efficient algorithm to compute $\Pi_{\mathcal{W}_0}\psi$ for any $\psi \in L^2(\Omega)$, where $\Pi_{\mathcal{W}_0} : L^2(\Omega) \to \mathcal{W}_0$ is an orthogonal projection operator (see section 5.3). In this paper, we always select $\mathcal{R}$ to be the indefinite integral operator in eq. (5.5).

5.2.4 Convergence theory for the preconditioned CG method

In this section, we show that the preconditioned CG method converges at a geometric rate when the operator preconditioner is bounded, is a smoother, and is a preconditioner for the Laplacian (see section 5.2.2). The standard bound on the convergence of the CG method for $Ax = b$ involves the condition number of A [110, Chap. 2]. Though this bound is not always descriptive, it is explicit and is the first canonical convergence result. Similarly, the convergence of our operator CG method can be bounded using the condition number of the operator $\mathcal{R}^*\mathcal{L}\mathcal{R}$ from a restricted subspace of $L^2(\Omega)$.

The condition number of $\mathcal{R}^*\mathcal{L}\mathcal{R} : \mathcal{W}_0 \to L^2(\Omega)$ is given by [88]:

$$\kappa_{\mathcal{W}_0}(\mathcal{R}^*\mathcal{L}\mathcal{R}) = \frac{\sup_{\phi\in\mathcal{W}_0,\|\phi\|=1} \mathcal{B}[\mathcal{R}\phi,\mathcal{R}\phi]}{\inf_{\phi\in\mathcal{W}_0,\|\phi\|=1} \mathcal{B}[\mathcal{R}\phi,\mathcal{R}\phi]}, \tag{5.7}$$

where $\mathcal{W}_0 = \{\phi \in L^2(\Omega) : \mathcal{R}\phi \in \mathcal{H}_0^1(\Omega)\}$. The following theorem bounds $\kappa_{\mathcal{W}_0}(\mathcal{R}^*\mathcal{L}\mathcal{R})$ and is used in corollary 5.2.2 to derive a CG convergence bound.

Theorem 5.2.2. *Let $\Omega = (-1, 1)$, $a, c \in L^\infty(\Omega)$, $a(x) > 0$ for $x \in \Omega$, $c(x) \geq 0$ for $x \in \Omega$, and $\mathcal{L}u = -(a(x)u'(x))' + c(x)u$ with bilinear form $\mathcal{B} : \mathcal{H}_0^1(\Omega) \times \mathcal{H}_0^1(\Omega) \to \mathbb{R}$. Given an operator preconditioner $\mathcal{R}$ that is bounded, is a smoother, and is a preconditioner for the Laplacian (see section 5.2.2), the (restricted) condition number of $\mathcal{R}^*\mathcal{L}\mathcal{R}$ is bounded. Furthermore,*

$$\kappa_{\mathcal{W}_0}(\mathcal{R}^*\mathcal{L}\mathcal{R}) \leq \frac{\gamma_1\|a\|_\infty + \|c\|_\infty\|\mathcal{R}\|_{op}^2}{\gamma_0 \inf_{x\in\Omega}|a(x)|},$$

where $\mathcal{W}_0 = \{\phi \in L^2(\Omega) : \mathcal{R}\phi \in \mathcal{H}_0^1(\Omega)\}$ and $\|\mathcal{R}\|_{op} = \sup_{\phi\in\mathcal{W}_0,\|\phi\|=1} \|\mathcal{R}\phi\|$.

Proof. If $\phi \in \mathcal{W}_0$, then $\mathcal{R}\phi \in \mathcal{H}_0^1(\Omega)$ and we have

$$\mathcal{B}[\mathcal{R}\phi,\mathcal{R}\phi] = \int_{-1}^{1} a(x)(\mathcal{R}\phi)'(x)(\mathcal{R}\phi)'(x)dx + \int_{-1}^{1} c(x)(\mathcal{R}\phi(x))^2 dx. \tag{5.8}$$

The first term in eq. (5.8) can be bounded as follows:

$$\int_{-1}^{1} a(x)(\mathcal{R}\phi)'(x)(\mathcal{R}\phi)'(x)dx \leq \|a\|_\infty\|(\mathcal{R}\phi)'\|^2 \leq \gamma_1\|a\|_\infty\|\phi\|^2,$$

where the last inequality uses the fact that $\mathcal{R}$ is a preconditioner for the Laplacian (see section 5.2.2). We also find that $\int_{-1}^{1} a(x)(\mathcal{R}\phi)'(x)(\mathcal{R}\phi)'(x)dx \geq \gamma_0 \inf_{x\in\Omega}|a(x)|\|\phi\|^2$. For the second term in eq. (5.8), we simply have

$$0 \leq \int_{-1}^{1} c(x)(\mathcal{R}\phi(x))^2 dx \leq \|c\|_\infty\|\mathcal{R}\phi\|^2 \leq \|c\|_\infty\|\mathcal{R}\|_{\text{op}}^2\|\phi\|^2.$$

The bound on $\kappa_{\mathcal{W}_0}(\mathcal{R}^*\mathcal{L}\mathcal{R})$ immediately follows. □

Similar statements to theorem 5.2.2 appear in the literature on operator preconditioners for Galerkin discretizations [72, 88]. Theorem 5.2.2 has a slightly different flavor because $\mathcal{R}$ and $\mathcal{L}$ are operators.

In eq. (5.6), the Krylov space is based on the operator $\mathcal{T} = \Pi^*_{\mathcal{W}_0} \mathcal{R}^* \mathcal{L} \mathcal{R} \Pi_{\mathcal{W}_0}$ and the (restricted) condition number of $\mathcal{T}$ immediately follows from theorem 5.2.2.

Corollary 5.2.1. *With the same assumptions as theorem 5.2.2, we have*

$$\kappa_{\mathcal{W}_0}(\mathcal{T}) = \kappa_{\mathcal{W}_0}(\mathcal{R}^* \mathcal{L} \mathcal{R}), \qquad \mathcal{T} = \Pi^*_{\mathcal{W}_0} \mathcal{R}^* \mathcal{L} \mathcal{R} \Pi_{\mathcal{W}_0},$$

where $\Pi_{\mathcal{W}_0} : L^2(\Omega) \to \mathcal{W}_0$ is the orthogonal projection operator onto $\mathcal{W}_0$.

The bound on $\kappa_{\mathcal{W}_0}(\mathcal{T})$ allows us to prove that $\|u - u_k\|_{\mathcal{L}}$ geometrically decays to zero as $k \to \infty$.

Corollary 5.2.2. *With the same assumptions as theorem 5.2.2, let $u_0 = 0, u_1, \ldots,$ be the CG iterates from algorithm 10. Then,*

$$\|u - u_k\|_{\mathcal{L}} \leq 2 \left(\frac{\sqrt{\kappa_{\mathcal{W}_0}(\mathcal{T})} - 1}{\sqrt{\kappa_{\mathcal{W}_0}(\mathcal{T})} + 1} \right)^k \|u\|_{\mathcal{L}}, , \qquad k \geq 0, \tag{5.9}$$

where $\mathcal{T} = \Pi^*_{\mathcal{W}_0} \mathcal{R}^* \mathcal{L} \mathcal{R} \Pi_{\mathcal{W}_0}$ and u is the exact solution to eq. (5.1)*.*

Proof. Corollary 5.2.1 shows that $\kappa_{\mathcal{W}_0}(\mathcal{T})$ is bounded. By copying the proof of the convergence bound for the CG method for matrices [110], we find that the iterates $v_0 = 0, v_1, v_2, \ldots,$ satisfy

$$\|v - v_k\|_{\mathcal{T}} \leq 2 \left(\frac{\sqrt{\kappa_{\mathcal{W}_0}(\mathcal{T})} - 1}{\sqrt{\kappa_{\mathcal{W}_0}(\mathcal{T})} + 1} \right)^k \|v\|_{\mathcal{T}}, \qquad k \geq 0,$$

where $u = \mathcal{R}v$. The result follows since $\|v\|^2_{\mathcal{T}} = \mathcal{B}[\mathcal{R}v, \mathcal{R}v] = \|\mathcal{R}v\|^2_{\mathcal{L}}$, and $u_k = \mathcal{R}v_k$. □

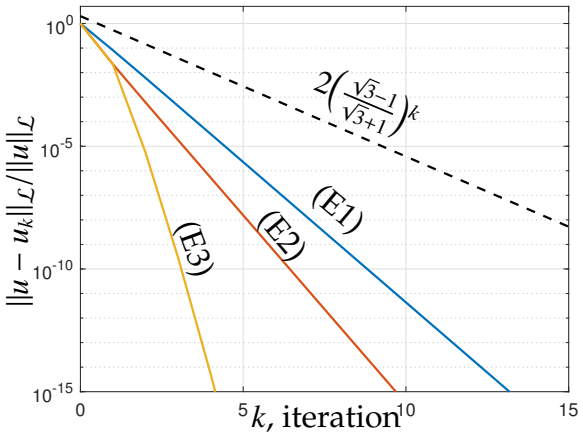


Figure 5.3: Convergence of the preconditioned CG method for three BVPs with zero Dirichlet boundary conditions. (E1): $-((2+\cos(\pi x))u')' = f$ (blue line), (E2): $-((1+x^2)u')' + (\frac{\pi}{4}\cos(\pi x))^2 u = f$ (red line), and (E3): $-u'' + 2(\frac{\pi}{4})^2 u = f$ (yellow line) with $f = (1+x^2)^{-1}$. Corollary 5.2.2 gives the same bound for these three examples (black dashed line). Note that $\mathcal{R}^* f \notin \mathcal{W}_0$ so an ancillary problem is solved before applying the CG method for these three BVPs (see section 5.2.5).

Corollary 5.2.2 implies that the preconditioned CG method in algorithm 10 constructs iterates $u_0 = 0, u_1, u_2, \ldots$, that converge geometrically to u in the $\|\cdot\|_{\mathcal{L}}$ norm. In other words, for an accuracy goal of $0 < \epsilon < 1$ we require

$$k \geq \left\lceil \frac{\log(2/\epsilon)}{\log\left(\sqrt{\kappa_{\mathcal{W}_0}(\mathcal{T})}+1\right) - \log\left(\sqrt{\kappa_{\mathcal{W}_0}(\mathcal{T})}-1\right)} \right\rceil,$$

iterations to guarantee that $\|u - u_k\|_{\mathcal{L}} \leq \epsilon \|u\|_{\mathcal{L}}$. Here, $\lceil x \rceil$ denotes the smallest integer greater than or equal to x. Since $\kappa_{\mathcal{W}_0}(\mathcal{T})$ is bounded, the preconditioned CG method can be terminated after a finite number of iterations.

Figure 5.3 shows the convergence of the preconditioned CG method compared to the error bound in eq. (5.9) when solving three BVPs using the indefinite integration preconditioner $\mathcal{R}v = \int_{-1}^{x} v(s)ds$. The convergence behavior of the preconditioned CG method comes with theoretical guarantees, and is a vast improvement over the convergence of the unpreconditioned CG method (see fig. 5.2 (right)).

5.2.5 General right-hand sides

Here, we remove the assumption that $\mathcal{R}^* f \in \mathcal{W}_0$ by solving an ancillary problem that converts f into a right-hand side that is amenable to our operator CG method.[7]

Write the solution to eq. (5.4) as $v = v_1 + v_2$, where v_2 is any solution from $\mathcal{W}_0$ that solves the following ancillary problem:

$$[\mathcal{R}\mathcal{R}^* \mathcal{L}\mathcal{R} v_2](\pm 1) = [\mathcal{R}\mathcal{R}^* f](\pm 1). \tag{5.10}$$

The remaining part of the solution, i.e., v_1, then satisfies

$$\mathcal{R}^* \mathcal{L}\mathcal{R} v_1 = \mathcal{R}^* g, \qquad g = (f - \mathcal{R}^* \mathcal{L}\mathcal{R} v_2).$$

By construction, $[\mathcal{R}\mathcal{R}^* g](\pm 1) = 0$ and since $g \in L^2(\Omega)$, we find that $\mathcal{R}^* g \in \mathcal{W}_0$. Therefore, we can solve

$$\mathcal{R}^* \mathcal{L}\mathcal{R} v_1 = \mathcal{R}^* g$$

via the preconditioned CG method in algorithm 10.

When $\mathcal{R}$ is the indefinite integral preconditioner in eq. (5.5), the condition at -1 in eq. (5.10) is trivially satisfied and the ancillary problem reduces to solving

$$\int_{-1}^{1} a(s) v_2(s) ds + \int_{-1}^{1} \left(\int_{s}^{1} c(t)(t+1) dt \right) v_2(s) ds = \int_{-1}^{1} (s+1) f(s) ds, \quad v_2 \in \mathcal{W}_0. \tag{5.11}$$

This problem can be solved efficiently by picking any $w \in \mathcal{W}_0$ such that the lefthand side of eq. (5.11), with v_2 replaced by w, is a scalar $\eta \neq 0$ and setting $v_2 = (\frac{1}{\eta} \int_{-1}^{1} (s+1) f(s) ds) w$. Usually, $w(s) = s$ is an adequate choice.

[7] In the standard Galerkin framework, one seeks to find a solution to eq. (5.1) via the weak formulation $\mathcal{B}[u, \psi] = \langle f, \psi \rangle$ for all $\psi \in \mathcal{H}_0^1(\Omega)$, even when $f \notin \mathcal{H}_0^1(\Omega)$. This is theoretically justified because $\mathcal{H}_0^1(\Omega)$ is a dense subspace of $L^2(\Omega)$. In our setup, the test space $\mathcal{W}_0$ is not a dense subset of $L^2(\Omega)$ so solving an ancillary problem is necessary.

5.3 Practical realizations of the operator CG method

We now describe two realizations of the theoretical framework in section 5.2 for solving eq. (5.1). While the theory in section 5.2 works for the solution space $\mathcal{H}_0^1(\Omega)$, in practice, we usually first define a dense subspace $\mathcal{V}$ of $\mathcal{H}^1(\Omega)$ and associated subspace $\mathcal{W} = \{w \in L^2(\Omega) : \mathcal{R}w \in \mathcal{V}\}$ on which the operations performed by the CG method can be efficiently computed. Provided that the operations performed by the CG method map functions from $\mathcal{W}$ to $\mathcal{W}$ and the right-hand side of eq. (5.1) and its variable coefficients are in $\mathcal{W}$, the preconditioned CG method in section 5.2 is unaware of the subspace $\mathcal{V}$. In this section, we consider: (1) $\mathcal{V}$ being the space of analytic functions and (2) $\mathcal{V}$ being the space of continuous piecewise analytic functions (with a finite number of fixed breakpoints). In these two cases the approximation space for the solution to eq. (5.1) is $\mathcal{V}_0 = \{\phi \in \mathcal{V} : \phi(\pm 1) = 0\} \subset \mathcal{H}_0^1(\Omega)$.

We have implemented (1) and (2) in Chebfun [42] in the `pcg` command, which follows the syntax of the standard MATLAB `pcg` command for matrices. Fortunately, object-oriented programming allows us to only have one implementation of the operator CG method for (1) and (2) as Chebfun automatically calls the appropriate underlying algorithms to compute inner-products, integrals, and derivatives via operator overloading. This is one of the advantages of developing a Krylov-based solver that works independently from the underlying discretization of the solution and right-hand side. Unlike most BVP solvers, our Krylov-based solvers have no fixed discretization. Instead, we let Chebfun automatically resolve the functions that appear during the operator CG method to machine precision [4]. A summary of the main operations that the preconditioned CG method requires is given in table 5.1, along with the corresponding

Table 5.1: Summary of the main operations that are required by the preconditioned CG method. The Chebfun commands that execute these mathematical operations are also given. Objected-oriented programming and operator overloading allows the same Chebfun command to employ different underlying algorithms depending on whether `p` and `q` are analytic or piecewise analytic.

Operation	Mathematical operation	Chebfun command
Preconditioner	$\int_{-1}^{x} p(s)ds$, $\int_{x}^{1} p(s)ds$	`cumsum(p)`, `sum(p)-cumsum(p)`
Differentiation	$p'(x)$	`diff(p)`
Product	$p(x)q(x)$	`p*q`
Inner-product	$\int_{-1}^{1} p(s)q(s)ds$	`p'*q`
Projector	$p - \frac{1}{2}\int_{-1}^{1} p(s)ds$	`p - mean(p)`

Chebfun commands.

5.3.1 Analytic functions

Let $\mathcal{V}$ be the space of functions that are analytic in an open neighborhood of $[-1, 1]$ and consider the preconditioner $\mathcal{R}\phi = \int_{-1}^{x} \phi(s)ds$. Note that the associated space $\mathcal{W}$ is closed under indefinite integration, differentiation, and function product, and that $\mathcal{R}$ is bounded, is a smoother, and preconditions the Laplacian. The choice of $\mathcal{V}$ and $\mathcal{R}$ completely determine a realization of the preconditioned CG method with the approximation space for the solution $\mathcal{V}_0 = \{\phi \in \mathcal{V} : \phi(\pm 1) = 0\}$. Here, we are implicitly assuming that the variable coefficients in eq. (5.1) are analytic functions or have been approximated by analytic functions.

In order to implement an efficient practical algorithm, we approximate analytic functions to within machine precision ϵ_{mach} by Chebyshev expansions. That is, for some integer $n \geq 0$ that is adaptively determined [4], we approximate an

analytic function $\phi \in \mathcal{V}$ by

$$\phi(x) \approx p(x) = \sum_{k=0}^{n} \alpha_k T_k(x), \qquad \|\phi - p\|_\infty < \epsilon_{\text{mach}} \|\phi\|_\infty, \tag{5.12}$$

where $T_k(x)$ is the degree k Chebyshev polynomial and $\|\cdot\|_\infty$ is the absolute maximum norm on $[-1, 1]$. If p is the Chebyshev interpolant of an analytic function ϕ, then the Chebyshev expansion coefficients in eq. (5.12) converge geometrically to zero [166, Chap. 8]. Moreover, the expansion coefficients $\{\alpha_k\}$ in eq. (5.12) can be computed in $O(n \log n)$ via the discrete Chebyshev transform [52]. To automatically resolve a function $\phi \in \mathcal{V}$ to machine precision, we call the Chebfun command `p = chebfun(phi)`.

There are a number of operations that the CG method must perform on the adaptively determined Chebyshev expansions:

Applying the preconditioner and its adjoint: For $p(x) = \sum_{k=0}^{n} \alpha_k T_k(x)$, we need to compute $\mathcal{R}p = \int_{-1}^{x} p(s)ds$. The Chebyshev expansion coefficients for $\mathcal{R}p$ can be computed by using a simple recurrence relation [109, Sec. 8.1], costing $O(n)$ operations. This is implemented in the Chebfun command `cumsum(p)`. Similarly, $\mathcal{R}^* p$ can be computed with the Chebfun command `sum(p)-cumsum(p)` in $O(n)$ operations.

Applying the differential operator: For $p(x) = \sum_{k=0}^{n} \alpha_k T_k(x)$, we need to compute $\mathcal{L}p$. If $\mathcal{L}p = -(a(x)p'(x))' + c(x)p(x)$ and $a(x)$ and $c(x)$ are analytic functions and represented by adaptively determined Chebyshev expansions, then we can compute $\mathcal{L}$ via the Chebun commands `Lp = -diff(a*diff(p))+c*p`. Computing the Chebyshev expansions of $p'(x)$ can be computed in $O(n)$ operations via a recurrence relation [109, p. 34] and the coefficients for $a(x)p(x)$ can be computed in $O(N \log N)$ oper-

ations with a discrete Chebyshev transform [52]. Here, N is the maximum polynomial degree required to resolve a and p.

Inner-products: Given $p(x) = \sum_{k=0}^{n} \alpha_k T_k(x)$ and $q(x) = \sum_{k=0}^{n} \beta_k T_k(x)$, we need to be able to compute

$$\langle p, q \rangle = \int_{-1}^{1} p(s)q(s)ds.$$

We compute this by Clenshaw–Curtis quadrature [166, Chap. 19], costing $O(n \log n)$ operations. The integral is computed by the Chebfun command `p'*q`.

Applying the projection operator: For $p(x) = \sum_{k=0}^{n} \alpha_k T_k(x)$, we need to compute the projection

$$\Pi_{w_0} p = p - \frac{1}{2} \int_{-1}^{1} p(s)ds.$$

This can be achieved in $O(n \log n)$ operations by using Clenshaw–Curtis quadrature for definite integration [166, Chap. 19]. The projection operator is computed by Chebfun with the command `p-mean(p)`.

Since this realization of the preconditioned CG method employs adaptively selected polynomials to resolve the solution of eq. (5.1), we compare our preconditioned CG method against adaptive implementations of the spectral collocation method[8] and the ultraspherical discretization [118]. Both these adaptive spectral methods are implemented in Chebfun.

To do the comparison, we consider the family of BVPs parametrized by ω_1 and ω_2 such that

$$-((2 + \cos(\omega_1 \pi x))u'(x))' = f(x) \text{ on } \Omega = (-1, 1), \qquad u(\pm 1) = 0,$$

[8]More precisely, we compare against rectangular spectral collocation [41], which performs a projection of the range of the matrices to automatically deal with boundary conditions of BVPs. Rectangular spectral collocation is employed by default in the `chebop` class of Chebfun [40].

where the right-hand side $f(x)$ is chosen so that $u(x) = \sin(\omega_2 \pi x)$ is the exact solution. We investigate two regimes: (a) ω_1 fixed, $\omega_2 \to \infty$ and (b) ω_2 fixed, $\omega_1 \to \infty$. In the first regime, a high degree polynomial is required to resolve the solution to machine precision while the variable coefficients of the BVP can be resolved by a low degree polynomial. This is a setting in which the ultraspherical spectral method is competitive with the preconditioned CG method (see fig. 5.4 (left)). In the second regime, the variable coefficients of the BVP require high degree polynomials to resolve, leading to dense spectral discretization matrices for both spectral collocation and the ultraspherical spectral method. In this setting, we find that it is computationally beneficial to employ our preconditioned CG method.

From these experiments and others, we learn that the preconditioned CG method is computationally beneficial compared to standard spectral methods employing direct solvers when spectral methods generate linear systems that are large and dense. A similar comparison can be made between direct and iterative solvers for linear systems.

5.3.2 Continuous functions that are piecewise analytic

Let $\mathcal{V} \subset \mathcal{H}^1(\Omega)$ be the space of continuous functions that are piecewise analytic with a finite number of fixed breakpoints $-1 = x_0 < x_1 < \cdots < x_{M+1} = 1$. That is, the space of continuous functions ϕ such that $\phi|_{[x_i, x_{i+1}]}$ is analytic in a neighborhood of $[x_i, x_{i+1}]$ for $0 \leq i \leq M$. Again, we take the preconditioner to be $\mathcal{R}\phi = \int_{-1}^{x} \phi(s)ds$. The induced space $\mathcal{W} = \{v \in L^2(\Omega) : \mathcal{R}v \in \mathcal{V}\}$ does not have a continuity requirement. The approximation space $\mathcal{W}$ is closed under indefinite

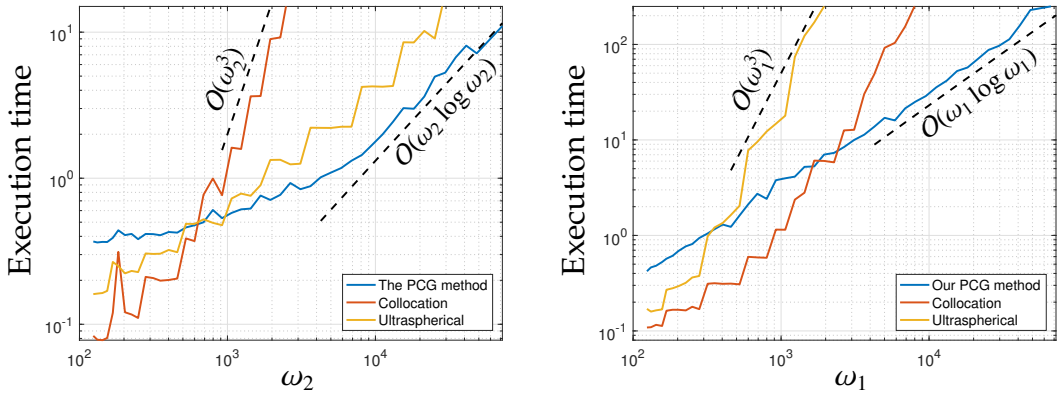


Figure 5.4: Comparison of execution timings of our preconditioned CG method (blue line), spectral collocation (red line), and the ultraspherical spectral method (yellow line) for the BVP $-((2+\cos(\omega_1 \pi x))u')' = f(x)$, $u(\pm 1) = 0$, where f is chosen so that $u(x) = \sin(\omega_2 \pi x)$ is the solution. All spectral methods are implemented in an adaptive manner to automatically resolve the BVP solution to essentially machine precision. The spectral collocation method and the ultraspherical spectral method discretizes the BVP and then solves the resulting linear system. Left: The parameter ω_2 is increased while $\omega_1 = 10$, which defines a family of BVPs for which the solution requires a high polynomial degree to resolve to machine precision. Right: The parameter ω_1 is increased while $\omega_2 = 10$, which defines a family of BVPs for which the variable coefficients require a high polynomial degree to resolve to machine precision. The polynomial degree required to resolve $\cos(\omega_1 \pi x)$ and $\sin(\omega_2 \pi x)$ on $[-1, 1]$ to machine precision is $O(\omega_1)$ and $O(\omega_2)$, respectively.

.

integration and multiplication and weak differentiation. This implies that all the functions that appear in the preconditioned CG method are in $\mathcal{W}$.

Given a function that is piecewise analytic, we represent it by subdividing the interval $[-1, 1]$ into $M + 1$ subintervals, i.e., $[-1, x_1] \cup [x_1, x_2] \cup \cdots \cup [x_M, 1]$, and representing the function by a Chebyshev expansion on each subinterval [123]. The Chebfun command that automatically determines the breakpoint locations and the polynomial degree to use on each subinterval is `p=chebfun(phi,'splitting','on')`. Any function that is computed during the CG method is automatically resolved in a piecewise fashion by Chebfun.

To solve for a piecewise smooth solution using spectral collocation or the ultraspherical spectral method, one has to construct a matrix that imposes the BVP operator on each subinterval along with continuity conditions at x_i for $1 \leq i \leq M$ [41]. In our preconditioned CG method the iterates v_k belong to $\mathcal{W}_0$, which is a space that contains functions that are not continuous. However, continuity on the approximate solutions $u_k = \mathcal{R}v_k$ is implicitly imposed because $\mathcal{R}$ acts as a smoother.

The algorithms to compute the tasks of applying the preconditioner, the differential operator, and the projection operator are almost immediate from the algorithms in section 5.3.1. For example, if $\phi \in \mathcal{V}$ and $x \in [x_m, x_{m+1}]$ for some $0 \leq m \leq M$, then

$$\mathcal{R}\phi = \int_{-1}^{x} \phi(s)ds = \sum_{i=0}^{m-1} \int_{x_i}^{x_{i+1}} \phi(s)ds + \int_{x_m}^{x} \phi(s)ds.$$

Therefore, to calculate the piecewise analytic function of $\mathcal{R}\phi$ on $[x_m, x_{m+1}]$ one performs indefinite integration on $[x_m, x_{m+1}]$ using a recurrence relation [109, Sec. 8.1] and adds to that the constant $\int_{-1}^{x_m} v(s)ds$ computed by applying Clenshaw–Curtis quadrature to each subinterval [166, Chap. 19].

Figure 5.5 demonstrates the preconditioned CG method on three BVPs with piecewise smooth variable coefficients. The solutions of which have the same breakpoints as the variable coefficients. Since Chebfun automatically determines breakpoint locations for piecewise smooth functions [123], our BVP solver automatically inherits this adaptivity. For piecewise continuous solutions we execute the same `pcg` command as in section 5.3.1 without modification. As can be seen from the convergence theory in section 5.2 and fig. 5.5 (right), the convergence rate of the CG method is independent of the smoothness of the solution.

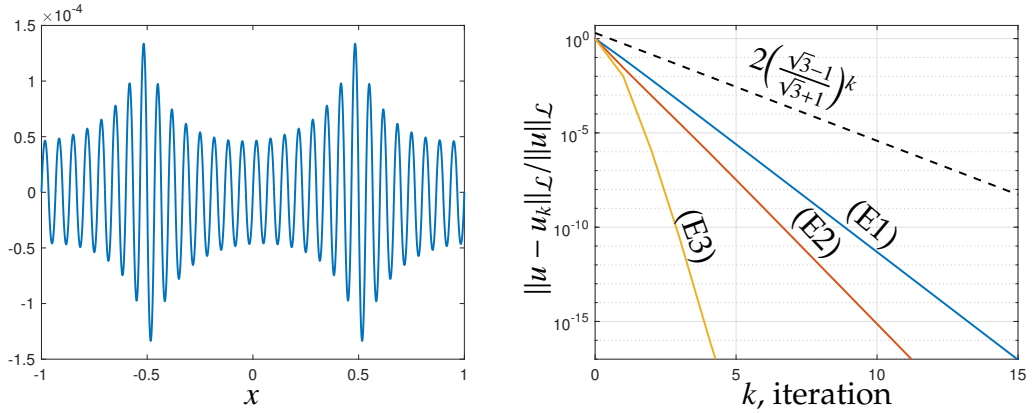


Figure 5.5: The preconditioned CG method for continuous functions that are piecewise analytic. Left: Solution to $-((1+2|\cos(\pi x)|)u')' = \text{sign}(\cos(30\pi x))$ with $u(\pm 1) = 0$. Right: The convergence of the preconditioned CG method for three BVPS with zero Dirichlet boundary conditions. (E1): $-((1+2|\cos(\pi x)|)u')' = f$ (blue line), (E2): $-((1+|\sin(\pi x^2)|)u')' + (\frac{\pi}{4})^2|\cos(2\pi x)|u = f$ (red line), and (E3): $-u'' + 2(\frac{\pi}{4})^2|\cos(20\pi x)|u = f$ (yellow line), where $f = (1+x^2)^{-1}$. Corollary 5.2.2 gives the same convergence bound for these three BVPs (black dashed line).

In fig. 5.6 we demonstrate the scaling of the PCG method on the family of BVPs with non-smooth coefficients:

$$-((2+\cos(\omega_2\pi x))u'(x))' + (2+|\cos(\omega_1\pi x^2)|)u(x) = f(x) \text{ on } \Omega = (-1,1), \quad u(\pm 1) = 0,$$

where $f(x)$ is chosen so that $u(x) = \sin(10\pi x)$ is the exact solution. We investigate two regimes: (1) $\omega_1 = \omega_2$, and (2) $\omega_2 = \omega_1^2$. In the first regime, the number of intervals increases but the degree of the polynomial on each subinterval stays roughly constant. In the second regime, both the number of intervals and the polynomial degree required to resolve the coefficients on each subinterval increases.

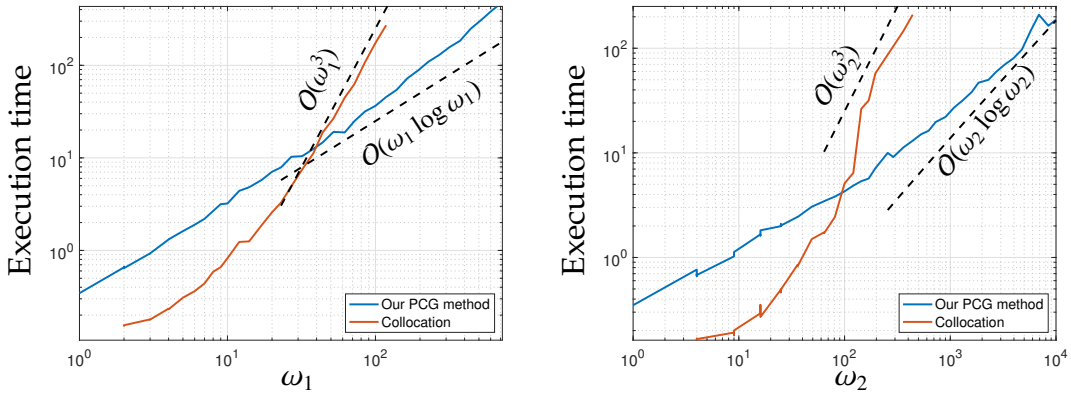


Figure 5.6: Comparison of execution timings of our preconditioned PCG method (blue line) and spectral collocation (red line) for the BVP $-((2 + \cos(\omega_2 \pi x))u'(x))' + (2 + |\cos(\omega_1 \pi x^2)|)u(x) = f(x)$, $u(\pm 1) = 0$, where f is chosen so that $u(x) = \sin(10\pi x)$ is the solution. All spectral methods are implemented in an adaptive manner to automatically resolve the BVP solution to essentially machine precision. Spectral collocation discretizes the BVP and then solves the resulting linear system. Left: The parameter ω_1 is increased and $\omega_2 = 10\omega_1$, which defines a family of BVPs for which the variable coefficients requires a high number of subintervals but a low polynomial degree on each subinterval to resolve to machine accuracy. Right: The parameter ω_1 is increased while $\omega_2 = \omega_1^2$, which defines a family of BVPs for which the variable coefficients require a high number of subintervals and a high polynomial degree on each subinterval to resolve the variable coefficients to machine accuracy. The number of subintervals required to resolve $|\cos(\omega_1 \pi x^2)|$ to machine precision is $O(\omega_1)$.

.

5.4 Other Krylov-based methods

The preconditioned CG method in section 5.2 has provided us with an operator analogue of a Krylov subspace for solving eq. (5.1) (see eq. (5.6)). Two additional Krylov subspace methods for solving $Ax = b$ are MINRES (for symmetric linear systems) [124] and GMRES (for general linear systems) [137]. In the matrix setting, MINRES and GMRES generate iterates by computing the best solution to $Ax = b$ from a Krylov subspace, as measured by the Euclidean norm of the

residual, i.e.,

$$x_k = \underset{y \in \mathcal{K}_k(A,b)}{\arg\min} \|b - Ay\|_2, \qquad \mathcal{K}_k(A,b) = \text{Span}\left\{b, Ab, \ldots, A^{k-1}b\right\}, \tag{5.13}$$

where $\|\cdot\|_2$ denotes the Euclidean norm of a vector.

Motivated by eq. (5.13), we set out to derive a MINRES and GMRES method for solving eq. (5.1) that constructs iterates so that

$$v_k = \underset{p \in \mathcal{K}_k(\mathcal{T}, \mathcal{R}^* f)}{\arg\min} \|\mathcal{R}^* f - \mathcal{T} p\|, \tag{5.14}$$

where $\mathcal{R}$ is given in eq. (5.5), and $\mathcal{T}$ and $\mathcal{K}_k(\mathcal{T}, \mathcal{R}^* f)$ are given in eq. (5.6). The hope is that the iterates $u_k = \mathcal{R} v_k$ converge to the solution u of eq. (5.1). In eq. (5.14), we assume that $\mathcal{R}^* f \in \mathcal{W}_0 = \{v \in L^2(\Omega) : \mathcal{R} v \in \mathcal{H}_0^1(\Omega)\}$; otherwise, the ancillary problem in section 5.2.5 is used to modify the right-hand side. In the case where $\mathcal{L}$ is self-adjoint with positive eigenvalues, a convergence bound analogous to eq. (5.9) holds for GMRES and MINRES. However, even though the preconditioned operator $\mathcal{R}^* \mathcal{L} \mathcal{R}$ is always bounded for the choice of the integration preconditioner, little can be said about the convergence of the methods in a general case.

5.4.1 The GMRES method for differential operators

The kth step of the GMRES method for solving $Ax = b$ computes an orthogonal basis for $\mathcal{K}_k(A, b)$ and then solves the least squares problem in eq. (5.13) for x_k. Analogously, our operator GMRES method computes an orthogonal basis for the Krylov subspace $\mathcal{K}_k(\mathcal{T}, \mathcal{R}^* f)$. The orthogonal basis is computed via the decomposition

$$\mathcal{T} \mathcal{Q}_k = \mathcal{Q}_{k+1} \tilde{H}_k, \tag{5.15}$$

where $\tilde{H}_k$ is a $(k+1) \times k$ upper Hessenberg matrix and $\mathcal{Q}_k$ is a quasimatrix with k orthonormal columns.[9] The decomposition is computed by an Arnoldi iteration on functions in $L^2(\Omega)$ using modified Gram–Schmidt (see algorithm 11).

Algorithm 11 Arnoldi iteration. Here, $\mathcal{T}$ is the operator in eq. (5.6) and $\mathcal{R}^* f \in \mathcal{W}_0$.

```
q_1 = R*f/||R*f||
for k = 2, ..., m do
    q_k = T q_{k-1}
    for j = 1, ..., k - 1 do
        h_{j,k-1} = <q_j, q_k>, q_k = q_k - h_{j,k-1} q_j
    end for
    h_{k,k-1} = ||q_k||, q_k = q_k / h_{k,k-1}
end for
```

Once an orthogonal basis for $\mathcal{K}_k(\mathcal{T}, \mathcal{R}^* f)$ is computed by algorithm 11, the iterates from eq. (5.14) can be computed as follows:

$$\underset{p \in \mathcal{K}_k(\mathcal{T}, \mathcal{R}^* f)}{\arg\min} \|\mathcal{R}^* f - \mathcal{T} p\| = \underset{y \in \mathbb{R}^k}{\arg\min} \|\mathcal{R}^* f - \mathcal{T} \mathcal{Q}_k y\| = \underset{y \in \mathbb{R}^k}{\arg\min} \left\| \|\mathcal{R}^* f\| e_1 - \tilde{H}_k y \right\|_2 ,$$

which is a standard least squares problem that is typically solved by updating a QR factorization of $\tilde{H}_k$ at each iteration using Givens rotations [171]. We derive the following operator GMRES method for eq. (5.1).

[9] A quasimatrix is a matrix whose columns are functions [156]. The quasimatrix has orthonormal columns if the columns are orthonormal with respect to the L^2 inner-product.

Algorithm 12 The preconditioned GMRES method for eq. (5.1), where $\mathcal{T}$ is the operator in eq. (5.6), $\mathcal{R}^* f \in \mathcal{W}_0$ and $0 < \epsilon < 1$ is a tolerance on the norm of the residual.

for $k = 1, 2, \ldots,$ **do**
  Compute Q_{k+1} and $\tilde{H}_k$ in eq. (5.15) using one step of algorithm 11
  Compute the QR factorization of $\tilde{H}_k$
  Solve $\rho = \min_y \left\| \|\mathcal{R}^* f\| e_1 - \tilde{H}_k y \right\|$
  if $\rho < \epsilon$ **then**
    $v = Q_k y$
    $u = \mathcal{R} v$
    stop iteration
  end if
end for

Unlike CG, the computational and storage costs of GMRES grows with the number of iterations. To avoid excessive storage costs, the GMRES method is usually restarted after m iterations for some integer m, i.e., v_m becomes an initial guess for a new GMRES method. The convergence behavior of the GMRES method is difficult to fully characterize and the statements that can be presented for convergence are analogous to those for the matrix GMRES method [171, Chap. 6]. Figure 5.7 (left) shows the convergence of the preconditioned GMRES on the BVP

$$-(e^x u')' + u' - 10u = \sin(30\pi x), \quad u(\pm 1) = 0$$

for different restarts. As observed in the matrix case the convergence can deteriorate with too frequent restarts, though iterates after restarting are computed more efficiently.

To study the scaling of the GMRES algorithm against other adaptive spectral methods, we consider the family of BVPs parametrized by ω_1 and ω_2 such that

$$-((2 + \cos(\omega_1 \pi x))u')' + (2 + \sin(\omega_1 \pi x))u' + u = f(x) \text{ on } \Omega = (-1, 1), \qquad u(\pm 1) = 0,$$

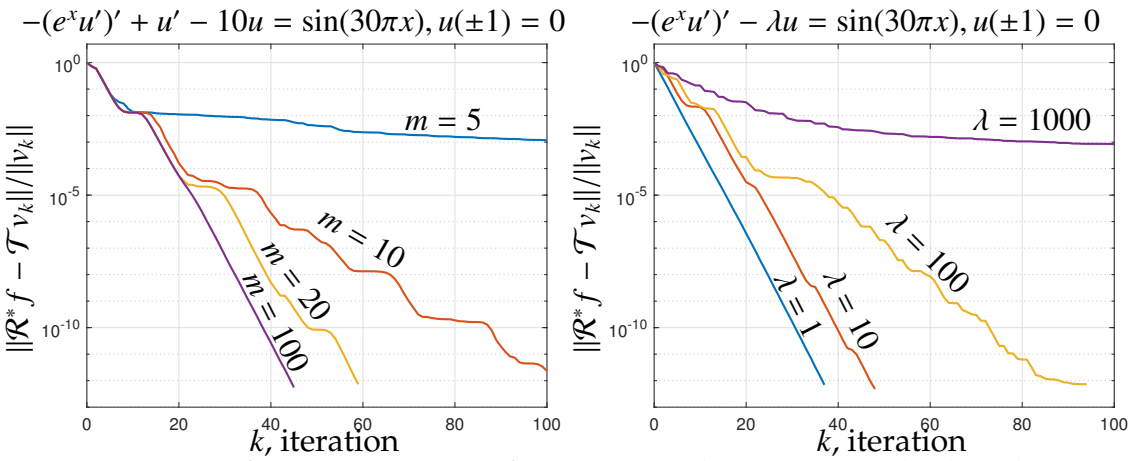


Figure 5.7: Left: Convergence of the restarted GMRES method with restarts every m iterations for $m = 5$ (blue line), $m = 10$ (red line), $m = 20$ (yellow line), and $m = 100$ (purple line). Right: Convergence of the MINRES method for $-(e^x u')' - \lambda u = \sin(30\pi x)$ with $u(\pm 1) = 0$ with $\lambda = 1$ (blue line), $\lambda = 10$ (red line), $\lambda = 100$ (yellow line), and $\lambda = 1000$ (purple line). The quality of the indefinite integral preconditioner in eq. (5.5) is reduced as λ increases.

where the right-hand side $f(x)$ is chosen so that $u(x) = \sin(\omega_2 \pi x)$ is the exact solution. Similarly to section 5.3.1, we investigate two regimes: (a) ω_1 fixed, $\omega_2 \to \infty$ and (b) ω_2 fixed, $\omega_1 \to \infty$. In the first regime, a high degree polynomial is required to resolve the solution to machine precision while the variable coefficients of the BVP can be resolved by a low degree polynomial. Figure 5.8 shows the result of the scaling study. We conclude that the GMRES method displays a similar scaling to the CG method and thus GMRES is also particularly beneficial over typical spectral methods when the variable coefficients of the BVP require high degree polynomials to resolve.

The operator GMRES method is implemented in Chebfun in the `gmres` command and has precisely the same realizations as the operator CG method (see section 5.3).

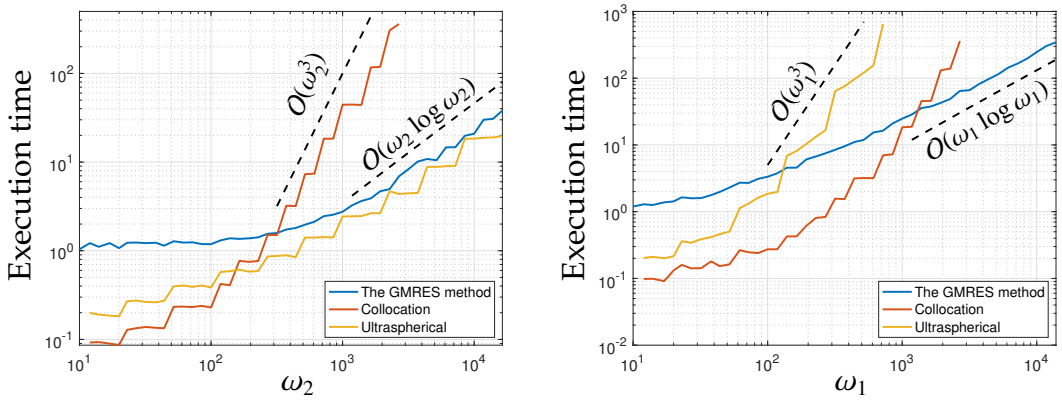


Figure 5.8: Comparison of execution timings of our preconditioned unrestarted GMRES method (blue line), spectral collocation (red line), and the ultraspherical spectral method (yellow line) for the BVP $-((2+\cos(\omega_1 \pi x))u')' + (2+\sin(\omega_1 \pi x))u' + u = f(x)$, $u(\pm 1) = 0$, where f is chosen so that $u(x) = \sin(\omega_2 \pi x)$ is the solution. All spectral methods are implemented in an adaptive manner to automatically resolve the BVP solution to essentially machine precision. Spectral collocation and the ultraspherical spectral method discretize the BVP and then solve the resulting linear system. Left: The parameter ω_2 is increased while $\omega_1 = 10$, which defines a family of BVPs for which the solution requires a high polynomial degree to resolve to machine precision. Right: The parameter ω_1 is increased while $\omega_2 = 10$, which defines a family of BVPs for which the variable coefficients require a high polynomial degree to resolve to machine precision. The polynomial degree required to resolve $\cos(\omega_1 \pi x)$ and $\sin(\omega_2 \pi x)$ on $[-1, 1]$ to machine precision is $O(\omega_1)$ and $O(\omega_2)$, respectively.

.

5.4.2 The MINRES method for differential operators

MINRES can be described as a special case of GMRES that applies when the linear system is symmetric. In that situation, the matrix $\tilde{H}_k$ reduces to a tridiagonal matrix and a Lanczos procedure is used instead of an Arnoldi iteration [124]. For self-adjoint second-order differential operators, it is analogous. Thus, our operator MINRES method is a GMRES method without restarts that exploits the fact that the operator is self-adjoint. An optimized implementation of MINRES notes that Q_k and $\mathcal{H}_k$ in eq. (5.15) do not need to be stored and that the solution y can be efficiently updated from previous iterates. The convergence properties

of operator MINRES are analogous to the convergence behavior of MINRES for solving linear systems.

We have implemented MINRES in Chebfun in the `minres` command, which has the same practical realizations as the CG method (see section 5.3). Figure 5.7 (right) shows the convergence of the preconditioned MINRES method on the family of BVPs $-(e^x u')' - \lambda u = \sin(30\pi x)$ with $u(\pm 1) = 0$ for different values of λ.

5.5 An extension to even-order BVPs

In this section, we describe the extension of continuous Krylov methods to even-order BVPs of the form:

$$\mathcal{L}u = f, \quad \Omega = (-1, 1), \qquad \frac{d^i u}{dx^i}(\pm 1) = 0, \quad 0 \le i \le K - 1, \tag{5.16}$$

where $\mathcal{L} : \mathcal{H}_0^K(\Omega) \cap \mathcal{H}^{2K}(\Omega) \to L^2(\Omega)$ and

$$\mathcal{L}u = \sum_{i=0}^{2K} a_i(x) \frac{d^i u}{dx^i}, \qquad a_{2K}(x) > 0. \tag{5.17}$$

Similarly to before, if $\mathcal{L}$ is self-adjoint with positive eigenvalues, then the CG method can be used whereas MINRES is for general self-adjoint operator and GMRES is for general operators. In this setting, our canonical preconditioner $\mathcal{R}$ is the integration preconditioner repeated K times, i.e.,

$$\mathcal{R}u(x) = \int_{-1}^{x} \int_{-1}^{x_1} \cdots \int_{-1}^{x_{K-1}} u(x_K) d_{x_K} \cdots dx_1, \tag{5.18}$$

$$\mathcal{R}^* u(x) = \int_{x}^{1} \int_{x_1}^{1} \cdots \int_{x_{K-1}}^{1} u(x_K) d_{x_K} \cdots dx_1. \tag{5.19}$$

If $\mathcal{L}$ can be written in the form $\mathcal{L}u = \sum_{i=0}^{K} (-1)^i \frac{d^i}{dx^i}\left(\hat{a}_i(x) \frac{d^i u}{dx^i}\right)$ with $\hat{a}_i(x) \ge 0$ for $0 \le i \le K$, then $\mathcal{L}$ is self-adjoint and has positive eigenvalues. In this case,

the continuous CG method described in Algorithm 10 can be employed without change and converges after a finite number of iteration. In particular, the condition number of the preconditioned operator can be bounded from above:

$$\kappa(\mathcal{R}^*\mathcal{L}\mathcal{R}) \leq \frac{\sum_{i=0}^{K} \|\hat{a}_i\|_\infty \left(\frac{\pi}{4}\right)^{2(K-i)}}{\inf_{x\in\Omega} |\hat{a}_K(x)|}. \tag{5.20}$$

The orthogonal projection $\Pi_{\mathcal{W}_0}$ onto the space $\mathcal{W}_0 = \{\phi \in L^2(\Omega) : \mathcal{R}\phi \in \mathcal{H}_0^K(\Omega)\}$ can also be easily expressed as

$$\Pi_{\mathcal{W}_0} u = u - p_K^{\text{best}}, \tag{5.21}$$

where p_K^{best} is the best polynomial of degree $\leq K$ to u in $L^2([-1,1])$. The polynomial p_K^{best} can also be simply computed by performing inner-products between u and the Legendre polynomials $P_i(x)$ for $0 \leq i \leq K$.

An auxiliary problem similar to the one in section 5.2.5 needs to be solved to ensure that the modified right-hand side g satisfies $\mathcal{R}^* g \in \mathcal{W}_0$. In this case, the auxiliary problem is to find a function find v_2 such that

$$\left[\frac{d^i}{dx^i}\left(\mathcal{R}\mathcal{R}^*\mathcal{L}\mathcal{R}v_2\right)\right](\pm 1) = \left[\frac{d^i}{dx^i}\left(\mathcal{R}\mathcal{R}^* f\right)\right](\pm 1) \tag{5.22}$$

$$\left[\frac{d^i}{dx^i}\left(\mathcal{R}v_2\right)\right](\pm 1) = 0 \tag{5.23}$$

holds for $0 \leq i \leq K-1$. These equations represent $4K$ discrete constraints and are solved via an $4K \times 4K$ linear system, finding that v_2 can be selected to be a polynomial of degree $\leq 4K-1$.

5.6 Extension to PDEs

Although the theory presented in section 5.2 extends to high dimensional problems, an implementation like the one presented in sections 5.3 and 5.4 is not straightforward. Indeed, in two dimensions, the solution of a differential equation with smooth coefficients is not necessarily smooth. For example, the solution to $\nabla^2 u = 1$ on $[-1, 1]^2$ with zero Dirichlet conditions has weak corner singularities. This means that even if we are able to approximate the coefficients of a PDE with a low degree polynomial, we may not be able to approximate the solution with a low degree polynomial. Therefore, looking for a practical iteration based on polynomial expansions which converges to the true solution in 2D is misguided as it would necessarily require the degree of iterates to explode as the number of iterations increases. In the face of this challenge, there are two possible ways forward:

- Look for the optimal solution over a finite-dimensional subspace by setting $\mathcal{V}_0$ to a finite dimensional subset of $\mathcal{H}_0^1(\Omega)$. This is similar to a typical spectral-Galerkin method [148].
- Choose a different basis to represent functions that are able to resolve weak corner singularities. For example, one could enrich the basis employed by a spectral method so that the output of a preconditioner built from the Laplacian can be adequately resolved (see, for example, [143]).

In what follows, we describe an implementation of the former. We highlight that this option is unsatisfying in this context, as we are giving up several attractive properties of the one-dimensional case: it does not converge to the true solution, and it is not adaptive. However, we show that this approach leads to

a competitive iterative solver for spectral discretization of PDEs.

We consider a partial differential equation

$$\mathcal{L}u(x,y) = f(x,y) \quad (x,y) \in [-1,1]^2$$
$$u(\pm 1, y) = u(x, \pm 1) = 0$$

where $\mathcal{L}$ is a self-adjoint uniformly elliptic operator, i.e.:

$$\mathcal{L}u = -\nabla \cdot (A(x,y)\nabla u(x,y)) + c(x,y)u(x,y)$$

with $[A(x,y)]_{i,j} = a_{ij}(x,y) \in L^\infty$ for $i,j \in \{1,2\}$, $c(x,y) \in L^\infty$, and $c(x,y) \geq 0$. The ellipticity assumptions implies that the spectrum of A is uniformly bounded [44] over $[-1,1]^2$ by, say, $0 < \lambda_{\min}(A) \leq \lambda_{\max}(A) < \infty$. We restrict ourselves to a finite dimensional subspace of $\mathcal{H}_0^1(\Omega)$ and set $\mathcal{V}_{0,n} = \mathcal{P}_{n,0}^2$ where

$$\mathcal{P}_{n,0}^2 := \{\phi(x,y) = \sum_{i=0}^{n}\sum_{j=0}^{n} \alpha_{i,j} x^i y^j \mid \phi(\pm 1, y) = \phi(x, \pm 1) = 0\}\,.$$

In order to implement an iteration similar to the 1D case, we need a preconditioner which is bounded, a smoother, and preconditions the Laplacian (see section 5.2.2). For this reason, we define the "square root" of the Laplacian (the analogue of the integration preconditioner for the one-dimensional case) with the formal relation:

$$-u_{xx} - u_{yy} = \left(\frac{\partial}{\partial x} - i\frac{\partial}{\partial y}\right)^* \left(\frac{\partial}{\partial x} - i\frac{\partial}{\partial y}\right) u$$

and set $\mathcal{R}u = \left(\frac{\partial}{\partial x} - i\frac{\partial}{\partial y}\right)^{-1} u$. This preconditioner defines a preconditioned set $\mathcal{W}_{0,n} = \{\phi \mid \mathcal{R}\phi \in \mathcal{P}_{n,0}^2\}$. In order to run a 2D CG method similar to the one in section 5.3, we need to be able to apply the composition of the operators $\mathcal{R}$ and $\Pi_{\mathcal{W}_{0,n}}$, where $\Pi_{\mathcal{W}_{0,n}}$ is the L^2 orthogonal projector onto $\mathcal{W}_{0,n}$. The operator $\Pi_{\mathcal{W}_{0,n}}\mathcal{R}$ may be written as:

$$(\Pi_{\mathcal{W}_0}\mathcal{R})\, u = \underset{v \in \mathcal{V}_{0,n}}{\arg\min} \left\| \left(\frac{\partial}{\partial x} - i\frac{\partial}{\partial y}\right) v - u \right\|_{L_2}^2 . \tag{5.24}$$

In section 5.6.1, we describe an algorithm to compute eq. (5.24) in $\mathcal{O}(n^2 \log(n))$ operations based on a Legendre polynomial basis and the technique presented in [47].

5.6.1 Computation of the 2D preconditioner

First, we write $v \in \mathcal{P}^2_{n,0}$ as $v = (1 - x^2)(1 - y^2)w$, where $w \in \mathcal{P}^2_{n-2}$. With this substitution, eq. (5.24) is equivalent to:

$$\min_{w \in \mathcal{P}_{n-2}} \left\| \left(\frac{\partial}{\partial x} - i \frac{\partial}{\partial y} \right) (1 - x^2)(1 - y^2)w - u \right\|^2_{L_2} .$$

The main idea of the algorithm is to compute a preconditioner by expanding the right-hand side w in an ultraspherical basis with parameter $\lambda = 3/2$ (denoted by $C^{(3/2)}$) [102, Tab. 18.3.1] and the solution u in a Legendre basis (denoted by P), and use recurrence relations between these two bases to obtain a sparse and structured representation of the operator that can be solved fast using the alternating direction implicit (ADI) method.

First, we use the relationship [102, (18.9.20)]:

$$\frac{d}{dx} \left(\left(1 - x^2 \right) C_n^{(3/2)}(x) \right) = -(n + 1)(n + 2) P_{n+1}(x) . \tag{5.25}$$

This relationship implies that the matrix representation of the operation $u \mapsto ((1 - x^2)u)''$ when the domain is represented in an $C^{(3/2)}$ basis and the output is

represented in a P basis is:

$$\hat{D} := \begin{bmatrix} 0 & & & & \\ -2 & & & & \\ & -6 & & & \\ & & \ddots & & \\ & & & -(n+1)(n+2) & 0 \end{bmatrix}.$$

Next, we use recurrence relation [102, (18.9.8)]:

$$(2n+3)\left(1-x^2\right) C_n^{3/2}(x) = -(n+1)(n+2) P_{n+2}(x) + (n+1)(n+2) P_n(x), \quad (5.26)$$

which implies that the matrix representation of the operation $u \mapsto (1-x^2)u$ when the domain is represented in an $C^{(3/2)}$ basis and the output is represented in a P basis has the form:

$$\hat{M} := \begin{bmatrix} \frac{1}{3} & & & & \\ 0 & \frac{6}{5} & & & \\ -\frac{1}{3} & 0 & \frac{12}{7} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{(n-1)n}{2n-1} & 0 & \frac{(n+1)(n+2)}{2n+3} \end{bmatrix}.$$

Using the normalized Legendre polynomials $\hat{P}_n := \frac{\sqrt{2n+1}}{2} P_n$, which are orthonormal with respect to the standard L^2 inner product, we find that the continuous minimization problem in eq. (5.24) reduces to a classical linear discrete least square problem. Thus, we diagonally scale the matrices $D := S\hat{D}$ and $M := S\hat{M}$ where S is the diagonal matrix of scaling factors $[S]_{i+1,i+1} = \sqrt{2/(2i+1)}$. Using this notation, and denoting the matrix of $C^{(3/2)}$ coefficients of w by $W \in \mathbb{C}^{n-2\times n-2}$ and the matrix of $\hat{P}$ coefficients by $u \in \mathbb{C}^{n\times n}$, eq. (5.24) is reduced to the following matrix linear equation:

$$\min_{W\in\mathbb{C}^{n-2\times n-2}} \|DUM^T - iMUD^T - W\|_F^2 = \min_{W\in\mathbb{C}^{n-2\times n-2}} \|A\mathrm{vec}(U) - \mathrm{vec}(W)\|_2^2 .$$

Here, $\mathrm{vec}(X) \in \mathbb{C}^{n^2}$ denotes the vector obtained by stacking the columns of $X \in \mathbb{C}^{n\times n}$, and $A := (D \otimes M) - i\,(M \otimes D)$. The normal equations of this least squares problem are:

$$A^* A \mathrm{vec}(U) = A^* \mathrm{vec}(W)\,.$$

Noting that D^*M is a skew-adjoint matrix, we find that

$$A^*A = (D^*D \otimes M^*M) + (M^*M \otimes D^*D)\,.$$

Let $R \in \mathbb{R}^{(n-2)\times(n-2)}$ be the square diagonal matrix that satisfies $RR = D^*D$, then

$$A^*A = (R \otimes R)\,[(I \otimes K) + (K \otimes I)]\,(R \otimes R)\ .$$

Here, $R \otimes R$ is a $(n-2)^2 \times (n-2)^2$ diagonal matrix. Therefore, it is trivial to solve linear systems involving $R \otimes R$ in $O(n^2)$. The matrix K is positive definite, banded, and $\kappa(K) = O(n^4)$, which means $[(I \otimes K) + (K \otimes I)]\,x = b$ can be solved in $O(n^2 \log(n) \log(1/\epsilon))$ operations using the ADI method with an accuracy tolerance of $0 < \epsilon < 1$ [47]. Thus, the total cost of solving eq. (5.24) is $O(n^2 \log(n))$ operations.

The remaining operations necessary to run the CG method can also be computed fast: differentiation in a Legendre basis costs $O(n^2)$ operations thanks to the sparse recurrence relations in [102, (18.9.19)] and products of Legendre series can be computed in $O(n^2 \log(n)^2)$ operations via a fast Legendre-to-Chebyshev transform [67]. A summary of all of the operations required to implement the 2D algorithm based on Legendre polynomials is given in table 5.2. The cost of one CG iteration is $O(n^2 \log(n)^2)$ operations.

The same argument as found in theorem 5.2.2 shows that

$$\kappa_{\mathcal{W}_{0,n}}(\mathcal{R}^* \mathcal{L} \mathcal{R}) \leq \frac{\lambda_{\max}(A) + \|c\|_\infty \|\mathcal{R}\|_{\mathrm{op}}^2}{\lambda_{\min}(A)}\ .$$

This implies that the number of iterations required to solve the system to some prescribed accuracy is independent of n. Consequently, the computational complexity of this spectral solver is $O(n^2 \log(n)^2)$ operations.

Table 5.2: Implementation and complexity of all operations required to implement the PCG method in 2D based on a Legendre basis.

Operations	Implementation	Complexity
Product	Legendre–Chebyshev transform [67] & fast Chebyshev product using FFT	$O(n^2 \log(n)^2)$
Differentiation	recurrence [102, (18.9.19)]	$O(n^2)$
Transform between $C^{(3/2)}$ and P	recurrence [102, (18.9.7)]	$O(n^2)$
Inner product	standard $\mathbb{R}^n$ inner product	$O(n^2)$
Preconditioner & projector	ADI [47]	$O(n^2 \log(n))$

We demonstrate the scaling of the algorithm and compare its running time to the two-dimensional extension of the ultraspherical spectral method [164]. We study two regimes: (1) The variable coefficients can be approximated with low degree polynomials, and (2) The variable coefficients require a high degree polynomials to be approximated. In the first regime, the ultraspherical spectral method produces sparse matrices and the resulting linear system can be solved by a direct solver in $O(n^4)$ operations. We also compare against a GMRES method preconditioned with an incomplete LU factorization (ILU(0)) [136, Chap. 10]. The execution time of the preconditioned GMRES method is observed to have complexity $O(n^{2.3})$. In the second regime, the matrix produced by the ultraspherical spectral method is dense and a direct solver requires $O(n^6)$ operations. In this case, we did not consider an iterative solver as the matrix is dense.

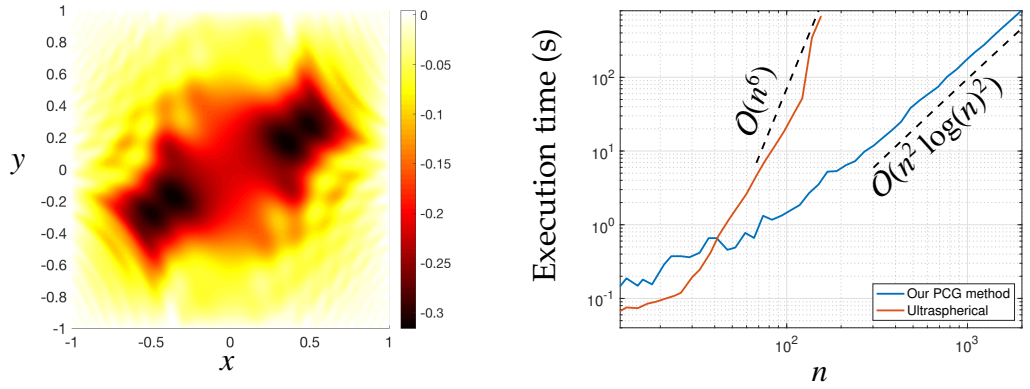


Figure 5.9: Numerical experiment on a PDE with variable coefficients that require high degree polynomials to approximate: $-\frac{\partial}{\partial x}((2+\sin(5xy\pi))\frac{\partial}{\partial x}u) - \frac{\partial}{\partial y}((2+\cos(50\pi x)\sin(10\pi y))\frac{\partial}{\partial y}u) = -100x\sin(20\pi x^2 y)\cos(4\pi(x+y))$ with zero Dirichlet conditions. Right: plot of the solution for $n = 1000$. Left: Comparison of execution timings of our PCG method and the ultrapspherical spectral method [164] for different values of n. The CG iteration was stopped when $\|r\|_2 < 10^{-13}$.

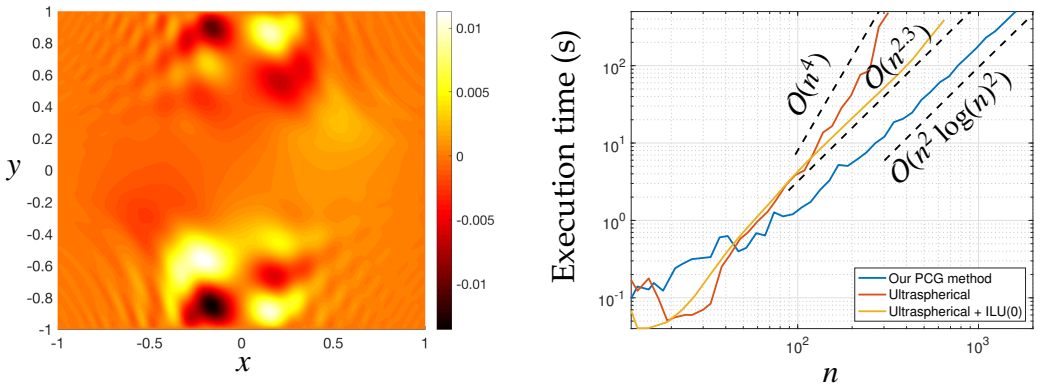


Figure 5.10: Numerical experiment on a PDE with variable coefficients that can be resolved with low degree polynomials: $-\frac{\partial}{\partial x}((2+\sin(\pi x)y^2)\frac{\partial}{\partial x}u) - \frac{\partial}{\partial y}((2+\cos(\pi x)\sin(\pi y))\frac{\partial}{\partial y}u) = 10y^2\sin(20\pi x^2 y)\cos(4\pi(x+y))$ with zero Dirichlet conditions. Left: plot of the solution for $n = 1000$. Right: Comparison of execution timings of our preconditioned CG method (blue line), the ultrapspherical spectral method using a sparse direct solver (the red line), and the ultrapspherical spectral method using GMRES preconditioned with ILU(0). The iterative methods were stopped when $\|r\|_2 < 10^{-13}$.

Conclusion

Operator analogues of the CG method, MINRES, and GMRES are derived for solving BVPs on $(-1, 1)$ that employ operator-function products. An operator preconditioner ensures that only a finite number of Krylov iterations are necessary to compute an approximate solution, and an orthogonal projection operator guarantees that the computed Krylov subspace imposes the boundary conditions of the BVP. The resulting iterative solvers are able to compute solutions from $\mathcal{H}_0^1(\Omega)$ and are competitive BVP solvers when a fast operator-function product is available.

BIBLIOGRAPHY

[1] J. Als-Nielsen and D. McMorrow. *Elements of Modern X-ray Physics*. Wiley, second edition, 2011.

[2] K. Alton and I. M. Mitchell. An ordered upwind method with precomputed stencil and monotone node acceptance for solving static convex Hamilton-Jacobi equations. *Journal of Scientific Computing*, 51(2):313–348, 2012.

[3] D. Attwood and A. Sakdinawat. *X-rays and Extreme Ultraviolet Radiation*. Cambridge University Press, second edition, 2017.

[4] J. L. Aurentz and L. N. Trefethen. Chopping a Chebyshev series. *ACM Transactions on Mathematical Software*, 43(4):33, 2017.

[5] O. Axelsson and J. Karátson. Equivalent operator preconditioning for elliptic problems. *Numerical Algorithms*, 50(3):297–380, 2009.

[6] M. Bardi and I. Capuzzo-Dolcetta. *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Springer Science & Business Media, 2008.

[7] B. R. Bean and E. Dutton. *Radio meteorology*. Dover Publications, 1966.

[8] A. Beck. *First-Order Methods in Optimization*, volume 25. SIAM, 2017.

[9] A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE transactions on image processing*, 18(11):2419–2434, 2009.

[10] J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2013.

[11] M. Born and E. Wolf. *Principles of Optics*. Cambridge University Press, seventh edition, 1999.

[12] P. Brucker. An $O(n)$ algorithm for quadratic knapsack problems. *Operations Research Letters*, 3(3):163–166, 1984.

[13] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods: Fundamentals in Single Domains*. Springer, 2010.

[14] R. Carmona and F. Delarue. *Probabilistic Theory of Mean Field Games with Applications I-II*. Springer, 2017.

[15] E. Cartee, L. Lai, Q. Song, and A. Vladimirsky. Time-dependent surveillance-evasion games. *preprint arXiv:1903.01332*, 2019.

[16] A. Chacon and A. Vladimirsky. Fast two-scale methods for eikonal equations. *SIAM Journal on Scientific Computing*, 34(2):A547–A578, 2012.

[17] A. Chacon and A. Vladimirsky. A parallel two-scale method for eikonal equations. *SIAM Journal on Scientific Computing*, 37(1):A156–A180, 2015.

[18] D. J. Ching and D. Gürsoy. Xdesign: an open-source software package for designing x-ray imaging phantoms and experiments. *Journal of Synchrotron Radiation*, 24(2):537–544, 2017.

[19] J. N. Clark, X. Huang, R. J. Harder, and I. K. Robinson. A continuous scanning mode for ptychography. *Optics Letters*, 39(20):6066–6069, Oct. 2014.

[20] J. N. Clark, X. Huang, R. J. Harder, and I. K. Robinson. Dynamic imaging using ptychography. *Physical Review Letters*, 112:113901, 2014.

[21] Z. Clawson, A. Chacon, and A. Vladimirsky. Causal domain restriction for eikonal equations. *SIAM Journal on Scientific Computing*, 36(5):A2478–A2505, 2014.

[22] Z. Clawson, X. Ding, B. Englot, T. A. Frewen, W. M. Sisson, and A. Vladimirsky. A bi-criteria path planning algorithm for robotics applications. *preprint arXiv:1511.01166*, 2015.

[23] J. M. Cowley and A. F. Moodie. Fourier images, I: the point source. *Proceedings of the Physical Society of London B*, 70(5):486–496, 1957.

[24] J. M. Cowley and A. F. Moodie. The scattering of electrons by atoms and crystals, I: a new theoretical approach. *Acta Crystallographica*, 10(10):609–619, Oct. 1957.

[25] M. G. Crandall and P.-L. Lions. Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, 277(1):1–42, 1983.

[26] R. A. Crowther, D. J. DeRosier, and A. Klug. The reconstruction of a three-dimensional structure from projections and its application to electron microscopy. *Proceedings of the Royal Society of London A*, 317(1530):319–340, June 1970.

[27] J. W. Daniel. The conjugate gradient method for linear and nonlinear operator equations. *SIAM Journal on Numerical Analysis*, 4(1):10–26, 1967.

[28] I. Das and J. E. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural optimization*, 14(1):63–69, 1997.

[29] D. Davis, B. Edmunds, and M. Udell. The sound of APALM clapping: faster nonsmooth nonconvex optimization with stochastic asynchronous PALM. In *Advances in Neural Information Processing Systems*, pages 226–234, 2016.

[30] J. Deng, Y. P. Hong, S. Chen, Y. S. G. Nashed, T. Peterka, A. J. F. Levi, J. Damoulakis, S. Saha, T. Eiles, and C. Jacobsen. Nanoscale x-ray imaging of circuit features without wafer etching. *Physical Review B*, 95(10):104111, 2017.

[31] J. Deng, Y. S. G. Nashed, S. Chen, N. W. Phillips, T. Peterka, R. Ross, S. Vogt, C. Jacobsen, and D. J. Vine. Continuous motion scan ptychography: characterization for increased speed in coherent x-ray imaging. *Optics Express*, 23(5):5438–5451, Feb. 2015.

[32] J. Deng, D. J. Vine, S. Chen, Q. Jin, Y. S. G. Nashed, T. Peterka, S. Vogt, and C. Jacobsen. X-ray ptychographic and fluorescence microscopy of frozen-hydrated cells using continuous scanning. *Scientific Reports*, 7(1):445, 2017.

[33] A. Desilles and H. Zidani. Pareto front characterization for multi-objective optimal control problems using Hamilton-Jacobi approach. *preprint*, 2018.

[34] M. Dierolf, A. Menzel, P. Thibault, P. Schneider, C. M. Kewish, R. Wepf, O. Bunk, and F. Pfeiffer. Ptychographic x-ray computed tomography at the nanoscale. *Nature*, 467(7314):436–439, Sept. 2010.

[35] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[36] J. Dobbie. Solution of some surveillance-evasion problems by the methods of differential games. In *Proceedings of the 4th International Conference on Operational Research, MIT, John Wiley and Sons, New York, New York*, 1966.

[37] G. Dockery. Modeling electromagnetic wave propagation in the troposphere using the parabolic equation. *IEEE Transactions on Antennas and Propagation*, 36(10):14641470, 1988.

[38] R. Douvenot, V. Fabbro, P. Gerstoft, C. Bourlier, and J. Saillard. A duct mapping method using least squares support vector machines. *Radio Science*, 43(6), 2008.

[39] R. Douvenot, V. Fabbro, P. Gerstoft, C. Bourlier, and J. Saillard. A duct mapping method using least squares support vector machines. *Radio Science*, 43(6), 2008.

[40] T. A. Driscoll, F. Bornemann, and L. N. Trefethen. The chebop system for automatic solution of differential equations. *BIT Numerical Mathematics*, 48(4):701–723, 2008.

[41] T. A. Driscoll and N. Hale. Rectangular spectral collocation. *IMA Journal of Numerical Analysis*, 36(1):108–132, 2015.

[42] T. A. Driscoll, N. Hale, and L. N. Trefethen. Chebfun Guide, 2014.

[43] M. Du and C. Jacobsen. Relative merits and limiting factors for x-ray and electron microscopy of thick, hydrated organic materials. *Ultramicroscopy*, 184:293–309, 2018.

[44] L. C. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, 2nd edition, 2010.

[45] M. Falcone and R. Ferretti. *Semi-Lagrangian approximation schemes for linear and Hamilton-Jacobi equations*, volume 133. SIAM, 2014.

[46] B. Fornberg. *A Practical Guide to Pseudospectral Methods*, volume 1. Cambridge university press, 1998.

[47] D. Fortunato and A. Townsend. Fast poisson solvers for spectral methods. *arXiv preprint arXiv:1710.11259*, 2017.

[48] V. Fountoulakis and C. Earls. Duct heights inferred from radar sea clutter using proper orthogonal bases. *Radio Science*, 2016.

[49] V. Fountoulakis and C. Earls. Inverting for maritime environments using proper orthogonal bases from sparsely sampled electromagnetic propagation data. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12):71667176, 2016.

[50] S. Gao, P. Wang, F. Zhang, G. T. Martinez, P. D. Nellist, X. Pan, and A. I. Kirkland. Electron ptychographic microscopy for three-dimensional imaging. *Nature Communications*, 8(1):1–8, July 2017.

[51] D. F. Gardner, M. Tanksalvala, E. R. Shanblatt, X. Zhang, B. R. Galloway, C. L. Porter, R. Karl Jr, C. Bevis, D. E. Adams, H. C. Kapteyn, et al. Subwavelength coherent imaging of periodic samples using a 13.5 nm tabletop high-harmonic light source. *Nature Photonics*, 11(4):259–263, 2017.

[52] W. M. Gentleman. Implementing Clenshaw-Curtis quadrature, II computing the cosine transformation. *Communications of the ACM*, 15(5):343–346, 1972.

[53] P. Gerstoft, D. F. Gingras, L. T. Rogers, and W. S. Hodgkiss. Estimation of radio refractivity structure using matched-field array processing. *IEEE Transactions on Antennas and Propagation*, 48(3):345–356, 2000.

[54] P. Gerstoft, L. T. Rogers, J. L. Krolik, and W. S. Hodgkiss. Inversion for refractivity parameters from radar sea clutter. *Radio science*, 38(3), 2003.

[55] D. F. Gingras, P. Gerstoft, and N. L. Gerr. Electromagnetic matched-field processing: Basic concepts and tropospheric simulations. *IEEE Transactions on Antennas and Propagation*, 45(10):1536–1545, 1997.

[56] T. M. Godden, R. Suman, M. J. Humphry, J. M. Rodenburg, and A. M. Maiden. Ptychographic microscope for three-dimensional imaging. *Optics Express*, 22(10):12513–12523, May 2014.

[57] R. M. Goldstein, H. A. Zebker, and C. L. Werner. Satellite radar interferometry: Two-dimensional phase unwrapping. *Radio Science*, 23(4):713–720, Dec. 2012.

[58] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins University Press, 1996.

[59] N. Gould, D. Orban, and T. Rees. Projected Krylov methods for saddle-point systems. *SIAM Journal on Matrix Analysis and Applications*, 35(4):1329–1343, 2014.

[60] O. Guéant, J.-M. Lasry, and P.-L. Lions. Mean field games and applications. In *Paris-Princeton lectures on mathematical finance 2010*, pages 205–266. Springer, 2011.

[61] A. Guigue. Approximation of the pareto optimal set for multiobjective optimal control problems using viability kernels. *ESAIM: COCV*, 20(1):95–115, 2014.

[62] M. Guizar-Sicairos, A. Diaz, M. Holler, M. S. Lucas, A. Menzel, R. A. Wepf, and O. Bunk. Phase tomography from x-ray coherent diffractive imaging projections. *Optics Express*, 19(22):21345–21357, Oct 2011.

[63] M. Guizar-Sicairos and J. R. Fienup. Phase retrieval with transverse translation diversity: a nonlinear optimization approach. *Optics Express*, 16(10):7264–7278, May 2008.

[64] D. Gürsoy. Direct coupling of tomography and ptychography. *Optics Letters*, 42(16):3169–4, 2017.

[65] D. Gürsoy, F. De Carlo, X. Xiao, and C. Jacobsen. TomoPy: a framework for the analysis of synchrotron tomographic data. *Journal of Synchrotron Radiation*, 21(5):1188–1193, 2014.

[66] T. Haack, C. Wang, S. Garrett, A. Glazer, J. Mailhot, and R. Marshall. Mesoscale modeling of boundary layer refractivity and atmospheric ducting. *Journal of Applied Meteorology and Climatology*, 49(12):24372457, 2010.

[67] N. Hale and A. Townsend. A fast, simple, and stable Chebyshev–Legendre transform using an asymptotic formula. *SIAM Journal on Scientific Computing*, 36(1):A148–A167, 2014.

[68] P. R. Halmos. *A Hilbert Space Problem Book*, volume 19. Springer Science & Business Media, 2012.

[69] B. L. Henke, E. M. Gullikson, and J. C. Davis. X-ray interactions: Photoabsorption, scattering, transmission, and reflection at E=50–30,000 eV, Z=1–92. *Atomic Data and Nuclear Data Tables*, 54:181–342, 1993.

[70] R. Hesse, D. R. Luke, S. Sabach, and M. K. Tam. Proximal heterogeneous block implicit-explicit method and application to blind ptychographic diffraction imaging. *SIAM Journal on Imaging Sciences*, 8(1):426–457, jan 2015.

[71] M. R. Hestenes and E. Stiefel. *Methods of conjugate gradients for solving linear systems*, volume 49. Journal of Research of the National Bureau of Standards, 1952.

[72] R. Hiptmair. Operator preconditioning. *Comput. Math. Appl.*, 52(5):699–706, 2006.

[73] M. Holler, A. Diaz, M. Guizar-Sicairos, P. Karvinen, E. Färm, E. Härkönen, M. Ritala, A. Menzel, J. Raabe, and O. Bunk. X-ray ptychographic computed tomography at 16 nm isotropic 3D resolution. *Scientific Reports*, 4:3857, 2014.

[74] M. Holler, M. Guizar-Sicairos, E. H. R. Tsai, R. Dinapoli, E. Müller, O. Bunk, J. Raabe, and G. Aeppli. High-resolution non-destructive three-dimensional imaging of integrated circuits. *Nature*, 543(7645):402–406, Mar. 2017.

[75] W. Hoppe. Beugung im inhomogenen Primarstrahlwellenfeld. I. Prinzip einer Phasenmessung von Elektronenbeungungsinterferenzen. *Acta Crystallographica Section A*, 25(4):495–501, 1969.

[76] X. Huang, K. Lauer, J. N. Clark, W. Xu, E. Nazaretski, R. Harder, I. K. Robinson, and Y. S. Chu. Fly-scan ptychography. *Scientific Reports*, 5:9074, Mar. 2015.

[77] X. Huang, H. Miao, J. Steinbrener, J. Nelson, D. Shapiro, A. Stewart, J. Turner, and C. Jacobsen. Signal-to-noise and radiation exposure considerations in conventional and diffraction x-ray microscopy. *Optics Express*, 17(16):13541–13553, July 2009.

[78] X. Huang, H. Yan, M. Ge, H. Öztürk, E. Nazaretski, I. K. Robinson, and Y. S. Chu. Artifact mitigation of ptychography integrated with on-the-fly scanning probe microscopy. *Applied Physics Letters*, 111(2):023103–5, July 2017.

[79] G. E. Ice, J. D. Budai, and J. W. Pang. The race to x-ray microbeam and nanobeam science. *Science*, 334:1234–1239, Dec. 2011.

[80] C. Jacobsen, J. Deng, and Y. Nashed. Strategies for high-throughput focused-beam ptychography. *Journal of Synchrotron Radiation*, 24(5):1078–1081, Sep 2017.

[81] F. B. Jensen. *Computational ocean acoustics*. American Inst. of Physics, 1994.

[82] U. S. Kamilov, I. N. Papadopoulos, M. H. Shoreh, A. Goy, C. Vonesch, M. Unser, and D. Psaltis. Learning approach to optical tomography. *Optica*, 2(6):517–6, 2015.

[83] U. S. Kamilov, I. N. Papadopoulos, M. H. Shoreh, A. Goy, C. Vonesch, M. Unser, and D. Psaltis. Optical tomographic image reconstruction based on beam propagation and sparse regularization. *IEEE Transactions on Computational Imaging*, 2(1):59–70, Feb. 2016.

[84] A. Karimian, C. Yardim, P. Gerstoft, W. S. Hodgkiss, and A. E. Barrios. Refractivity estimation from sea clutter: An invited review. *Radio Science*, 46(6), 2011.

[85] K. Katayama, Y. Takeda, K. Kuwabara, and S. Kuwahara. A novel photocatalytic microreactor bundle that does not require an electric power source. *Chemical Communications*, 48(59):7368, 2012.

[86] L. Kaufman. Eigenvalue problems in fiber optic design. *SIAM Journal on Matrix Analysis and Applications*, 28(1):105117, 2006.

[87] D. E. Kerr. *Propagation of short radio waves*. McGraw-Hill, 1951.

[88] R. C. Kirby. From functional analysis to iterative methods. *SIAM Review*, 52(2):269–293, 2010.

[89] A. Kumar and A. Vladimirsky. An efficient method for multiobjective optimal control and optimal control subject to integral constraints. *Journal of Computational Mathematics*, pages 517–551, 2010.

[90] C. Larabell and M. Le Gros. X-ray tomography generates 3-D reconstructions of the yeast, *Saccharomyces cerevisiae*, at 60-nm resolution. *Molecular Biology of the Cell*, 15:957–962, 2004.

[91] G. Lefurjah, R. Marshall, T. Casey, T. Haack, and D. D. F. Boyer. Synthesis of mesoscale numerical weather prediction and empirical site-specific radar clutter models. *IET Radar, Sonar & Navigation*, 4(6):747, 2010.

[92] R. J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-state and Time-dependent Problems*, volume 98. SIAM, 2007.

[93] J. Lewin. *Decoy in pursuit-evasion games*. PhD thesis, Department of Aeronautics and Astronautics, Stanford University, 1973.

[94] J. Lewin and J. Breakwell. The surveillance-evasion game of degree. *Journal of Optimization Theory and Applications*, 16(3-4):339–353, 1975.

[95] J. Lewin and G. Olsder. Conic surveillance evasion. *Journal of Optimization Theory and Applications*, 27(1):107–125, 1979.

[96] K. Li, M. Wojcik, and C. Jacobsen. Multislice does it all—calculating the performance of nanofocusing x-ray optics. *Optics Express*, 25(3):1831–16, 2017.

[97] P. Li and A. M. Maiden. Multi-slice ptychographic tomography. *Scientific Reports*, 8:2049, Jan. 2018.

[98] J. Liesen and Z. Strakoš. *Krylov Subspace Methods: Principles and Analysis*. Oxford University Press, 2013.

[99] J. Lim, A. Goy, M. H. Shoreh, M. Unser, and D. Psaltis. Learning tomography assessed using Mie theory. *Physical Review Applied*, 9(3):034027, 2018.

[100] E. H. Linfoot and E. Wolf. Diffraction images in systems with an annular aperture. *Proceedings of the Physical Society of London B*, 66(398):145–149, 1953.

[101] A. R. Lowry, C. Rocken, S. V. Sokolovskiy, and K. D. Anderson. Vertical profiling of atmospheric refractivity from ground-based GPS. *Radio Science*, 37(3), 2002.

[102] D. W. Lozier. NIST digital library of mathematical functions. *Annals of Mathematics and Artificial Intelligence*, 38(1-3):105–119, 2003.

[103] A. M. Maiden, M. J. Humphry, and J. M. Rodenburg. Ptychographic transmission microscopy in three dimensions using a multi-slice approach. *Journal of the Optical Society of America A*, 29(8):1606–1614, Aug. 2012.

[104] A. M. Maiden, M. J. Humphry, M. C. Sarahan, B. Kraus, and J. M. Rodenburg. An annealing algorithm to correct positioning errors in ptychography. *Ultramicroscopy*, 120(C):64–72, Sept. 2012.

[105] A. M. Maiden and J. M. Rodenburg. An improved ptychographical phase retrieval algorithm for diffractive imaging. *Ultramicroscopy*, 109(10):1256–1262, Aug. 2009.

[106] J. Málek and Z. Strakoš. *Preconditioning and the Conjugate Gradient Method in the Context of Solving PDEs*. SIAM, 2014.

[107] R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.

[108] J. Maser and G. Schmahl. Coupled wave description of the diffraction by zone plates with high aspect ratios. *Optics Communications*, 89:355–362, 1992.

[109] J. C. Mason and D. C. Handscomb. *Chebyshev polynomials*. CRC Press, 2002.

[110] G. Meurant. *The Lanczos and conjugate gradient algorithms: from theory to finite precision computations*, volume 19. SIAM, 2006.

[111] H. Mimura, S. Handa, T. Kimura, H. Yumoto, D. Yamakawa, H. Yokoyama, S. Matsuyama, K. Inagaki, K. Yamamura, Y. Sano, K. Tamasaku, Y. Nishino, M. Yabashi, T. Ishikawa, and K. Yamauchi. Breaking the 10 nm barrier in hard-x-ray focusing. *Nature Physics*, 6(2):122–125, 2010.

[112] J.-M. Mirebeau. Efficient fast marching with Finsler metrics. *Numerische mathematik*, 126(3):515–557, 2014.

[113] I. M. Mitchell and S. Sastry. Continuous path planning with multiple constraints. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 5, pages 5502–5507. IEEE, 2003.

[114] Y. S. Nashed, D. J. Vine, T. Peterka, J. Deng, R. Ross, and C. Jacobsen. Parallel ptychographic reconstruction. *Optics Express*, 22(26):32082–32097, 2014.

[115] Y. S. G. Nashed, T. Peterka, J. Deng, and C. Jacobsen. Distributed automatic differentiation for ptychography. *Procedia Computer Science*, 108:404–414, 2017.

[116] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer, 2006.

[117] S. Olver. GMRES for the differentiation operator. *SIAM Journal on Numerical Analysis*, 47(5):3359–3373, 2009.

[118] S. Olver and A. Townsend. A fast and well-conditioned spectral method. *SIAM Review*, 55(3):462–489, 2013.

[119] E. L. Ortiz. The tau method. *SIAM Journal on Numerical Analysis*, 6(3):480–492, 1969.

[120] M. J. Osborne and A. Rubinstein. *A course in game theory*. MIT press, 1994.

[121] O. Ozgun, G. Apaydin, M. Kuzuoglu, and L. Sevgi. PETOOL: Matlab-based one-way and two-way split-step parabolic equation tool for radiowave propagation over variable terrain. *Computer Physics Communications*, 182(12):26382654, 2011.

[122] H. Öztürk, H. Yan, Y. He, M. Ge, Z. Dong, M. Lin, E. Nazaretski, I. K. Robinson, Y. S. Chu, and X. Huang. Multi-slice ptychography with large numerical aperture multilayer Laue lenses. *Optica*, 5(5):601–607, May 2018.

[123] R. Pachón, R. B. Platte, and L. N. Trefethen. Piecewise-smooth chebfuns. *IMA Journal of Numerical Analysis*, 30(4):898–916, 2009.

[124] C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 12(4):617–629, 1975.

[125] P. M. Pelz, M. Guizar-Sicairos, P. Thibault, I. Johnson, M. Holler, and A. Menzel. On-the-fly scans for x-ray ptychography. *Applied Physics Letters*, 105:251101, Dec. 2014.

[126] S. Penton and E. Hackett. Rough ocean surface effects on evaporative duct atmospheric refractivity inversions using genetic algorithms. *Radio Science*, 53(6):804–819, 2018.

[127] T. Pock and S. Sabach. Inertial proximal alternating linearized minimization (iPALM) for nonconvex and nonsmooth problems. *SIAM Journal on Imaging Sciences*, 9(4):1756–1787, 2016.

[128] J. Pozderac, J. Johnson, C. Yardim, C. Merrill, T. de Paolo, E. Terrill, F. Ryan, and P. Frederickson. *x*-band beacon-receiver array evaporation

duct height estimation. *IEEE Transactions on Antennas and Propagation*, 66(5):2545–2556, 2018.

[129] D. Qi and A. Vladimirsky. Corner cases, singularities, and dynamic factoring. *Journal of Scientific Computing*, 79(3):1456–1476, 2019.

[130] T. Raghavan. Zero-sum two-person games. *Handbook of game theory with economic applications*, 2:735–768, 1994.

[131] J. M. Rodenburg and H. M. Faulkner. A phase retrieval algorithm for shifting illumination. *Applied Physics Letters*, 85(20):4795–4797, 2004.

[132] L. T. Rogers. Demonstration of an efficient boundary layer parameterization for unbiased propagation estimation. *Radio Science*, 33(6):1599–1608, 1998.

[133] J. R. Rowland and S. M. Babin. Fine-scale measurements of microwave refractivity profiles with helicopter and low-cost rocket probes. *Johns Hopkins APL Technical Digest*, 8(4):413–417, 1987.

[134] F. J. Ryan. *User's Guide for the VTRPE (Variable Terrain Radio Parabolic Equation) Computer Model*. 1991.

[135] B. P. Rynne and M. A. Youngson. *Linear Functional Analysis*. Springer Science & Business Media, 2000.

[136] Y. Saad. *Iterative methods for sparse linear systems*, volume 82. SIAM, 2003.

[137] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific Computing*, 7(3):856–869, 1986.

[138] A. E. Sakdinawat and D. T. Attwood. Nanoscale x-ray imaging. *Nature Photonics*, 4(12):840–848, Dec. 2010.

[139] G. Schneider. Zone plates with high efficiency in high orders of diffraction described by dynamical theory. *Applied Physics Letters*, 71(16):2242–2244, 1997.

[140] G. Schneider, S. Rehbein, and S. Werner. Volume effects in zone plates. In A. Erko, M. Idir, T. Krist, and A. G. Michette, editors, *Modern Developments in X-ray and Neutron Optics*, pages 137–171. Springer, 2008.

[141] T. Schoonjans, A. Brunetti, B. Golosio, M. S. del Rio, V. A. Solé, C. Ferrero, and L. Vincze. The xraylib library for x-ray–matter interactions. recent developments. *Spectrochimica Acta Part B: Atomic Spectroscopy*, 66(11-12):776–784, Nov. 2011.

[142] A. Schropp and C. G. Schroer. Dose requirements for resolving a given feature in an object by coherent x-ray diffraction imaging. *New Journal of Physics*, 12(3):035016, Mar. 2010.

[143] W. W. Schultz, N. Lee, and J. P. Boyd. Chebyshev pseudospectral method of viscous flows with corner singularities. *Journal of scientific computing*, 4(1):1–24, 1989.

[144] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[145] J. A. Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, volume 3. Cambridge university press, 1999.

[146] J. A. Sethian and A. Vladimirsky. Ordered upwind methods for static Hamilton–Jacobi equations: Theory and algorithms. *SIAM Journal on Numerical Analysis*, 41(1):325–363, 2003.

[147] D. A. Shapiro, Y.-S. Yu, T. Tyliszczak, J. Cabana, R. Celestre, W. Chao, K. Kaznatcheev, A. L. D. Kilcoyne, F. Maia, S. Marchesini, Y. S. Meng, T. Warwick, L. L. Yang, and H. A. Padmore. Chemical composition mapping with nanometre resolution by soft x-ray microscopy. *Nature Photonics*, 8(10):765–769, 2014.

[148] J. Shen. Efficient spectral-Galerkin method I. direct solvers of second- and fourth-order equations using Legendre polynomials. *SIAM Journal on Scientific Computing*, 15(6):1489–1505, 1994.

[149] J. Shen, T. Tang, and L.-L. Wang. *Spectral Methods: Algorithms, Analysis and Applications*, volume 41. Springer Science & Business Media, 2011.

[150] K. Shimomura, A. Suzuki, M. Hirose, and Y. Takahashi. Precession x-ray ptychography with multislice approach. *Physical Review B*, 91(21):214114, June 2015.

[151] C.-W. Shu. High order numerical methods for time dependent Hamilton-Jacobi equations. In *Mathematics and computation in imaging science and information processing*, pages 47–91. World Scientific, 2007.

[152] T. D. Sikora and S. Ufermann. *Marine Atmospheric Boundary Layer Cellular Convection and Longitudinal Roll Vortices*. NOAA, 2004.

[153] I. Sirkova. Brief review on PE method application to propagation channel modeling in sea environment. *Open Engineering*, 2(1), Jan 2012.

[154] H. Soner. Optimal control with state-space constraint. I. *SIAM Journal on Control and Optimization*, 24(3):552–561, 1986.

[155] J. C. H. Spence, U. Weierstall, and M. Howells. Phase recovery and lensless imaging by iterative methods in optical, x-ray and electron diffrac-

tion. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 360(1794):875–895, May 2002.

[156] G. W. Stewart. *Afternotes goes to graduate school: lectures on advanced numerical analysis*. SIAM, 1998.

[157] A. Suzuki, S. Furutaku, K. Shimomura, K. Yamauchi, Y. Kohmura, T. Ishikawa, and Y. Takahashi. High-resolution multislice x-ray ptychography of extended thick objects. *Physical Review Letters*, 112(5):053903, Feb. 2014.

[158] Y. Takahashi, Y. Nishino, R. Tsutsumi, H. Kubo, H. Furukawa, H. Mimura, S. Matsuyama, N. Zettsu, E. Matsubara, T. Ishikawa, and K. Yamauchi. High-resolution diffraction microscopy using the plane-wave field of a nearly diffraction limited focused x-ray beam. *Physical Review B*, 80(5):054103, 2009.

[159] R. Takei, W. Chen, Z. Clawson, S. Kirov, and A. Vladimirsky. Optimal control with budget constraints and resets. *SIAM Journal on Control and Optimization*, 53(2):712–744, 2015.

[160] A. Tarantola. *Inverse problem theory and methods for model parameter estimation*, volume 89. SIAM, 2005.

[161] P. Thibault, M. Dierolf, A. Menzel, O. Bunk, C. David, and F. Pfeiffer. High-resolution scanning x-ray diffraction microscopy. *Science*, 321(5887):379, 2008.

[162] P. Thibault and A. Menzel. Reconstructing state mixtures from diffraction measurements. *Nature*, 494(7435):68–71, Feb. 2013.

[163] S. Tijs. *Semi-infinite linear programs and semi-infinite matrix games*. Katholieke Universiteit Nijmegen. Mathematisch Instituut, 1976.

[164] A. Townsend and S. Olver. The automatic solution of partial differential equations using a global spectral method. *Journal of Computational Physics*, 299:106–123, 2015.

[165] L. N. Trefethen. *Spectral Methods in MATLAB*. SIAM, 2000.

[166] L. N. Trefethen. *Approximation Theory and Approximation Practice*, volume 128. SIAM, 2013.

[167] L. N. Trefethen and D. Bau III. *Numerical Linear Algebra*. SIAM, 1997.

[168] E. H. R. Tsai, I. Usov, A. Diaz, A. Menzel, and M. Guizar-Sicairos. X-ray ptychography with extended depth of field. *Optics Express*, 24(25):29089, 2016.

[169] Y.-H. R. Tsai, L.-T. Cheng, S. Osher, and H.-K. Zhao. Fast sweeping algorithms for a class of Hamilton–Jacobi equations. *SIAM Journal on Numerical Analysis*, 41(2):673–694, 2003.

[170] J. N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40(9):1528–1538, 1995.

[171] H. A. Van der Vorst. *Iterative Krylov Methods for Large Linear Systems*, volume 13. Cambridge University Press, 2003.

[172] S. Vasudevan, R. H. Anderson, S. Kraut, P. Gerstoft, L. T. Rogers, and J. L. Krolik. Recursive Bayesian electromagnetic refractivity estimation from radar sea clutter. *Radio Science*, 42(2), 2007.

[173] M. Wagner, P. Gerstoft, and T. Rogers. Estimating refractivity from propagation loss in turbulent media. *Radio Science*, 51(12):1876–1894, 2016.

[174] W. Wang and M. A. Carreira-Perpinán. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. *arXiv preprint arXiv:1309.1541*, 2013.

[175] Y. Wang, C. Jacobsen, J. Maser, and A. Osanna. Soft x-ray microscopy with a cryo scanning transmission x-ray microscope: II. Tomography. *Journal of Microscopy*, 197(1):80–93, Jan. 2000.

[176] J. A. C. Weideman and L. N. Trefethen. The eigenvalues of second-order spectral differentiation matrices. *SIAM Journal on Numerical Analysis*, 25(6):1279–1298, 1988.

[177] G. J. Williams, M. Pfeifer, I. Vartanyants, and I. K. Robinson. Effectiveness of iterative algorithms in recovering phase in the presence of noise. *Acta Crystallographica A*, 63(1):36–42, Dec. 2006.

[178] A. Willitsford and C. Philbrick. Lidar description of the evaporative duct in ocean environments. In *Remote Sensing of the Coastal Oceanic Environment*, volume 5885, page 58850G. International Society for Optics and Photonics, 2005.

[179] F. Xie, S. Syndergaard, E. R. Kursinski, and B. M. Herman. An approach for retrieving marine boundary layer refractivity from GPS occultation data in the presence of superrefraction. *Journal of Atmospheric and Oceanic Technology*, 23(12):1629–1644, 2006.

[180] H. Yan, J. Maser, A. Macrander, Q. Shen, S. Vogt, G. Stephenson, and H. Kang. Takagi-Taupin description of x-ray dynamical diffraction from diffractive optics with large numerical aperture. *Physical Review B*, 76(11):115438, 2007.

[181] C. Yardim, P. Gerstoft, and W. Hodgkiss. Estimation of radio refractivity from radar clutter using Bayesian Monte Carlo analysis. *IEEE Transactions on Antennas and Propagation*, 54(4):13181327, 2006.

[182] C. Yardim, P. Gerstoft, and W. S. Hodgkiss. Atmospheric refractivity tracking from radar clutter using Kalman and particle filters. *2007 IEEE Radar Conference*, 2007.

[183] F. Zhang, G. Pedrini, and W. Osten. Phase retrieval of arbitrary complex-valued fields through aperture-plane modulation. *Physical Review A*, 75(4):043805, 2007.

[184] F. Zhang, I. Peterson, J. Vila-Comamala, A. Diaz, F. Berenguer, R. Bean, B. Chen, A. Menzel, I. K. Robinson, and J. M. Rodenburg. Translation position determination in ptychographic coherent diffraction imaging. *Optics Express*, 21(11):13592, 2013.

[185] Q. Zhang and K. Yang. Study on evaporation duct estimation from point-to-point propagation measurements. *IET Science, Measurement & Technology*, 12(4):456–460, 2018.

[186] H. Zhao. A fast sweeping method for eikonal equations. *Mathematics of computation*, 74(250):603–627, 2005.

[187] X. Zhao. Evaporation duct height estimation and source localization from field measurements at an array of radio receivers. *IEEE Transactions on Antennas and Propagation*, 60(2):1020–1025, 2012.

[188] X.-F. Zhao, S.-X. Huang, and H.-D. Du. Theoretical analysis and numerical experiments of variational adjoint approach for refractivity estimation. *Radio Science*, 46(1), 2011.